

Parallel and Explicit Time Integration of Transient Nonlinear Magneto- and Electroquasistatic Fields

Dissertation

zur Erlangung des akademischen Grades eines
Doktor der Ingenieurwissenschaften (Dr.-Ing.)

der Fakultät für Elektro- Informations- und Medientechnik
der Bergischen Universität Wuppertal vorgelegt

von

Jennifer Susanne Dutiné, M.Sc.,
geboren am 29.06.1990 in Berlin

Referent: Prof. Dr. rer. nat. Markus Clemens

Koreferent: Prof. Dr. rer. nat. Sebastian Schöps

Tag der Einreichung: 08.04.2020

Tag der mündlichen Prüfung: 08.07.2020

The PhD thesis can be quoted as follows:

urn:nbn:de:hbz:468-20200812-083618-4

[<http://nbn-resolving.de/urn/resolver.pl?urn=urn%3Anbn%3Ade%3Ahbz%3A468-20200812-083618-4>]

DOI: 10.25926/qvyh-3g26

[<https://doi.org/10.25926/qvyh-3g26>]

Parallel and Explicit Time Integration of Transient Nonlinear Magneto- and Electroquasistatic Fields

Dissertation

zur Erlangung des akademischen Grades eines Doktor der Ingenieurwissenschaften
(Dr.-Ing.)

vorgelegt von

Jennifer Susanne Dutiné, M.Sc.

Contents

List of Figures	iii
List of Tables	vii
List of Algorithms	ix
List of Abbreviations	xi
List of Symbols	xiii
1 Introduction	1
1.1 Previous and Related Works	4
1.2 Overview	7
2 Mathematical Description of Electromagnetic Field Problems	11
2.1 Maxwell's Equations	11
2.2 The Magnetoquasistatic and Electroquasistatic Approximation	13
3 Discretization of Partial Differential Equations in Space and Time	15
3.1 The Finite Element Method	15
3.1.1 FEM applied to the PDE of Magnetoquasistatic Field Problems	16
3.1.2 FEM Applied to the PDE of Electroquasistatic Field Problems .	18
3.2 Time Integration Methods	20
3.2.1 The Implicit Euler Method	20
3.2.2 The Explicit Euler Method	22
4 Solving Large Systems of Algebraic Equations	25
4.1 The Conjugate Gradient Method	25
4.2 The Sparse LU-Decomposition	27

5 (Semi-) Explicit Time Integration of the Magnetoquasistatic Field Problem	29
5.1 Application of the Moore-Penrose Pseudo-Inverse	29
5.1.1 Numerical Validation: Application of a Pseudo-Inverse	35
5.1.1.1 Results for the TEAM 7 benchmark problem	35
5.1.1.2 Results for the TEAM 10 benchmark problem	41
5.2 Improved Start Vector Generation Within Semi-Explicit Time Stepping	49
5.2.1 Subspace Projection Extrapolation and Cascaded Subspace Pro- jection Extrapolation	50
5.2.2 Proper Orthogonal Decomposition for MRHS Problems	52
5.2.3 Numerical Validation: Improved Start Vectors	54
5.3 Selective Matrix Update Bypass Strategy	65
5.3.1 Numerical Validation of the Selective Matrix Update Bypass Strategy	66
5.4 GPU-Accelerated Inverse Computations	75
5.4.1 Numerical Validation	76
6 Parallel-In-Time Integration of Transient Electroquasistatic Field Prob- lems Using Parareal	81
6.1 Mathematical Formulation	81
6.2 Numerical Validation	84
6.2.1 Results of Simulations with a Linear Field Grading Material	87
6.2.2 Results of Simulations with a Nonlinear Field Grading Material	91
7 Conclusion and Outlook	97
Bibliography	101
Publications by the Author	113

List of Figures

3.1	Stability region of the implicit Euler method in the z -plane, as e.g. in [45]. The region of stability is colored in blue.	21
3.2	Stability region of the explicit Euler method in the z -plane, as e.g. in [45]. The region of stability is colored in blue.	23
5.1	The computational domain Ω and its boundary $\partial\Omega$ comprise three sub-domains: The nonconducting region (Ω_n), which has a constant permeability, the conducting region (Ω_c), where eddy currents can be excited and the permeability might be nonlinear and a nonconducting region (Ω_{sc}), where no eddy currents are excited and the excitation is placed. .	31
5.2	Model geometry of the TEAM 7 benchmark problem	36
5.3	Excitation current flowing in the source coil of the TEAM 7 benchmark problem	37
5.4	Meshes of the TEAM 7 benchmark problem	38
5.5	Results for the magnitude of the B_z -component of the magnetic flux density along the evaluation line P computed using the explicit or the implicit Euler method, in comparison with the measurement results published in [36]. The results are obtained using the fine mesh resulting in 247,276 degrees of freedom.	39
5.6	Field plots for the TEAM 7 benchmark problem	40
5.7	Model geometry of the TEAM 10 benchmark problem.	42
5.8	Current flowing in the exciting source coil of the TEAM 10 benchmark problem. The current is described with the function $i_S = (1 - \exp(-t/\tau))$, where $\tau = 0.05$ s.	43
5.9	Ferromagnetic nonlinear B - H -curve of the steel plates according to the material data published in [70].	44
5.10	Meshes for the TEAM 10 benchmark problem in chapter 5.1.1.2	45

5.11	Field plots of the eddy currents for the TEAM 10 benchmark problem in chapter 5.1.1.2	46
5.12	Field plots of the magnetic flux density for the TEAM 10 benchmark problem in chapter 5.1.1.2	47
5.13	Results on the coarse grid for the magnitude of the z-component of the magnetic flux density evaluated along line S1.	48
5.14	Meshes for the TEAM 10 benchmark problem in chapter 5.2.3	56
5.15	Comparison of simulation results evaluated on a fine mesh of 700,000 dofs and the measurement results published in [70] along lines S1, S2, and S3.	57
5.16	Field plots of the magnetic flux density for the TEAM 10 benchmark problem in chapter 5.2.3	58
5.17	Simulation results obtained using the implicit Euler method and the explicit Euler method with varying CG tolerance evaluated along lines S1, S2, and S3.	59
5.18	Averagely required number of PCG iterations when applying the CSPE method for computing improved start vectors for the PCG method.	60
5.19	Maximum required number of column vectors in the operator \mathbf{V}_K of the CSPE method.	61
5.20	Resulting simulation times when applying the CSPE method for computing improved start vectors for the PCG method.	62
5.21	Averagely required number of PCG iterations using CSPE, or POD, or the old solution vector from the previous time step as start vector for the PCG method.	63
5.22	Simulation times using POD, CSPE, or the old solution vector from the previous time step as start vector for the PCG method.	63
5.23	The first two periods of a 50 Hz ramped sinusoidal current, referred to as "ramped sinusoidal current".	67
5.24	The first two periods of a 50 Hz ramped sinusoidal current, referred to as "ramped sinusoidal current".	68
5.25	Number of updates for the matrix \mathbf{K}_{cc} and different tolerances, if the "switch-on current" is the excitation current.	69
5.26	Simulation times T_S for the evaluation of the matrix \mathbf{K}_{cc} and all tolerances chosen for the "switch-on current" excitation.	70

5.27	Resulting number of updates for the matrix \mathbf{K}_{cc} and all tolerances chosen, if the "ramped sinusoidal current" is the excitation current.	71
5.28	Simulation times T_S for the update strategy for the matrix \mathbf{K}_{cc} and all tolerances for the "ramped sinusoidal current" excitation current.	72
5.29	Relative error in the average flux density computed according to equation (5.28), comparing the selective update strategy vs. the full evaluation strategy.	73
5.30	Deviation of the results computed for the average magnetic flux density along line S1 using the implicit Euler method and various time step sizes for time integration from the results obtained using explicit time integration and updating the matrix \mathbf{K}_{cc} in every time step.	74
5.31	Mesh of the TEAM 10 problem resulting in 50,035 degrees of freedom.	77
5.32	Results for the magnitude of average magnetic flux density on the coarse mesh evaluated along line S3.	78
5.33	Results for the magnitude of average magnetic flux density on the fine mesh evaluated along line S3.	79
5.34	Simulation times on the coarse mesh.	80
5.35	Simulation times on the finer mesh.	80
6.1	An exemplary plot of the solution on the coarse and the fine time grid is shown at one iteration of the Parareal algorithm. The coarse time grid consists of six time intervals with a time step size Δt_c . The solution for x on the coarse time grid is shown in green. The solution on the fine time grid with a time step size Δt_f is shown in each coarse time interval in blue. The exact solution, against which the Parareal algorithm's solution will converge, is shown in pink.	82
6.2	Geometry of the two-dimensional model of an IEC 60099-4 surge arrester. The air region is colored in turquoise. The components of the surge arrester are colored in red, green, and orange. Areas, where the surge arrester pierces through the air plane are colored in white.	85
6.3	A triangular mesh resulting in 23,577 degrees of freedom is applied to the model geometry shown in Figure 6.2.	86
6.4	Results for the electric potential of the mesh resulting in 23,577 degrees of freedom with linear field grading material.	88

6.5	Results for the electric potential of the mesh resulting in 51,528 degrees of freedom with linear field grading material.	88
6.6	Results for the voltage stress of the mesh resulting in 23,577 degrees of freedom with linear field grading material.	89
6.7	Results for the voltage stress of the mesh resulting in 51,528 degrees of freedom with linear field grading material.	89
6.8	Real simulation times of the problem with 23,577 degrees of freedom and with linear field grading material.	90
6.9	Real simulation times of the problem with 51,528 degrees of freedom and with linear field grading material.	90
6.10	Conductivity curve of the nonlinear field grading material of the surge arrester used.	92
6.11	Real simulation times for the mesh resulting in 23,577 degrees of freedom with a nonlinear field grading material.	94
6.12	Real simulation times for the mesh resulting in 23,577 degrees of freedom with a nonlinear field grading material, without the simulation time without ImIm-Parareal.	94
6.13	Results for the electric potential of the mesh resulting in 23,577 degrees of freedom with a nonlinear field grading material.	95
6.14	Results for the voltage stress of the mesh resulting in 23,577 degrees of freedom with a nonlinear field grading material.	95

List of Tables

5.1	Time step sizes and resulting simulation times for all simulations of the TEAM 7 benchmark problem carried out. All simulations are run on a server with an Intel Xeon E5-2660 processor.	41
5.2	Time step sizes and resulting simulation times for all simulations of the TEAM 10 benchmark problem carried out. All simulations are run on a server with an Intel Xeon E5-2660 processor.	47
6.1	Time step lengths for simulations with linear field grading material using the implicit and the explicit Euler method for the sequential time integration. .	87
6.2	Time step lengths for simulations with linear field grading material using the Parareal algorithm.	87
6.3	Time step lengths for simulations with a nonlinear field grading material using the implicit and the explicit Euler method for the sequential time integration.	93
6.4	Time step lengths for simulations with a nonlinear field grading material using the Parareal algorithm.	93

List of Algorithms

1	Solve $\mathbf{Ax} = \mathbf{b}$ using the Conjugate Gradient Method	26
2	Parareal algorithm, as given in [94, 48]	83

List of Abbreviations

- ACA** adaptive cross approximation
- AMG** algebraic multigrid
- BEM** boundary element method
- CFL** Courant-Friedrichs-Lewy criterion
- CG** conjugate gradient method
- CPU** central processing unit
- CSPE** cascaded subspace projection extrapolation
- DAE** differential algebraic equation
- DG-FEM** discontinuous Galerkin finite element method
- DOF** degree of freedom
- EQS** electroquasistatic
- FDTD** finite difference time domain method
- FEM** finite element method
- FIT** finite integration technique
- GPGPU** general purpose graphics processing unit
- GPU** graphics processing unit
- ICCG** incomplete Cholesky conjugate gradient
- IEC** international electrotechnical commission

- LU** LU-decomposition (decomposition of a matrix into a lower (L) and upper (U) triangular matrix)
- MEQSICO** magneto-electro-quasistatic simulation code
- MRHS** multiple right-hand side
- MQS** magnetoquasistatic
- ODE** ordinary differential equation
- PCG** preconditioned conjugate gradient
- PDE** partial differential equation
- PML** perfectly matched layer
- POD** proper orthogonal decomposition
- RAM** random access memory
- SDIRK** singly diagonal implicit Runge-Kutta method
- SPE** subspace projection extrapolation
- SSOR** symmetric successive over-relaxation
- SVD** singular value decomposition
- TEAM** testing electromagnetic analysis methods

List of Symbols

Symbols in Chapter 2

∇	Nabla operator
\vec{A}	magnetic vector potential
\vec{B}	magnetic flux density
\vec{D}	electric flux density
\vec{E}	electric field strength
ε	permittivity
ε_0	electric constant
ε_r	relative permittivity
\vec{H}	magnetic field strength
I	time interval
\vec{J}	electric current density
\vec{J}_S	impressed current density
κ	electric conductivity
μ	permeability
μ_0	magnetic constant
μ_r	relative permeability
ν	magnetic reluctivity
\vec{r}	position vector
ϱ	electric charge density
t	time
t_0	start point in time of a time interval
t_{end}	end point in time of a time interval
ϕ	electric scalar potential
Ω	spatial computational domain

\vec{P}_E	electric polarization vector
\vec{P}_M	magnetic polarization vector
W_E	electric energy
W_M	electric energy

Symbols in Chapter 3.1

∇	Nabla operator
a	bilinear form
a_i	degree of freedom
b	bilinear form
\vec{A}	magnetic vector potential
Γ	boundary of the spatial computational domain Ω
ε	permittivity
E_{ij}	Edge connecting nodes i and j
$H(\text{curl}, \Omega)$	Sobolev space
$H_0(\text{curl}, \Omega)$	Sobolev space
$H^1(\Omega)$	Sobolev space
i	index, corresponding to node numbers in the mesh
j	index, corresponding to node numbers in the mesh
\vec{J}_S	source current density
κ	electric conductivity
k	index, corresponding to node numbers in the mesh
\mathbf{K}	stiffness matrix
\mathbf{K}_D	stiffness matrix for Dirichlet values
l	index, corresponding to node numbers in the mesh
$L^2(\Omega)$	space of square-integrable functions
\mathbf{M}	mass matrix
\mathbf{M}_D	mass matrix for Dirichlet values
ν	magnetic reluctivity
\vec{n}	outward pointing normal vector of the boundary Γ
n_e	total number of edges in the mesh
n_n	total number of nodes in the mesh

N_i	nodal basis function with index i
N_j	nodal basis function with index j
\vec{v}	vector test function
v	scalar test function
ϕ	electric scalar potential
Ω	spatial computational domain
r	linear form
\mathbf{r}	right-hand side vector
\vec{r}	position vector
t	time
\vec{t}_{ij}	unit vector pointing along edge E_{ij}
u_j	degree of freedom
$\vec{w}_{i,j}^{(1)}$	1st order edge element basis function
\mathbf{x}	vector containing degrees of freedom
\mathbf{x}_D	vector for Dirichlet values

Symbols in Chapter 3.2

f	function
i	index
\mathbf{J}	Jacobian matrix
\mathbf{K}	stiffness matrix
l	index
\mathbf{M}	mass matrix
\mathbf{r}	right-hand side vector
\mathbf{r}_i	right-hand side vector at the i -th point in time
t	time
t_i	i -th point in time
Δt	time step size
u	index
v	index
\mathbf{x}	vector containing degrees of freedom
\mathbf{x}_i	vector containing degrees of freedom at the i -th point in time

y continuous function
 y_0 initial value

Symbols in Chapter 4

∇ Nabla operator
 α_k scalar coefficient in the CG method
 \mathbf{A} system matrix
 β_k scalar coefficient in the CG method
 \mathbf{b} right-hand side vector
 \mathbf{d} direction vector
 f function
 \mathbf{h} vector using in the PCG method
 i index
 j index
 k index
 \mathbf{L} lower triangular matrix
 n number of rows in the system matrix \mathbf{A}
 \mathbf{P} Preconditioning matrix
 \mathbf{Q} Decomposition of the Preconditioning matrix \mathbf{P}
 \mathbf{r} residual vector
 tol tolerance
 \mathbf{U} upper triangular matrix
 V subspace
 \mathbf{x} solution vector
 \mathbf{x}_0 initial solution vector
 \mathbf{y} vector used in the LU-decomposition
 \mathbf{z} vector used in the CG method

Symbols in Chapter 5

\vec{A}	magnetic vector potential
\mathbf{a}	vector of degrees of freedom representing the magnetic vector potential
\mathbf{a}_c	vector of degrees of freedom in the conductive subdomain Ω_c
\mathbf{a}_n	vector of degrees of freedom in the nonconductive subdomains Ω_n and Ω_{sc}
\mathbf{a}_p	general representation of the matrix vector product $\mathbf{K}_{nn}^+ \mathbf{r}_p$
$\mathbf{a}_{p0,POD}$	improved start vector when evaluating \mathbf{K}_{nn}^+ , computed using POD
\vec{B}	magnetic flux density
\mathbf{B}_{evr}	average magnetic flux density computed at position S1 updating \mathbf{K}_{cc} in every time step
\mathbf{B}_{impl}	average magnetic flux density evaluated using the implicit Euler scheme
\mathbf{B}_{sel}	average magnetic flux density computed at position S1 using the selective update strategy
B_z	magnitude of the z-component of the magnetic flux density
d_z	air gap width
ϵ	offset parameter
ϵ_{CG}	tolerance
ϵ_{POD}	threshold value
H	magnitude of the magnetic field strength
h	smallest edge length in the mesh
I	source current
\mathbf{I}	identity matrix
i	index
i_S	time dependent source current
j	index
\mathbf{r}_p	general representation of the vectors \mathbf{K}_{nn}^+ is multiplied with
\mathbf{j}_S	transient source current density discretized in space
$\mathbf{j}_{S,n}$	transient source current density in the nonconductive subdomain Ω_{sc}
κ	electric conductivity

List of Symbols

\mathbf{K}	stiffness matrix of reluctivities, i.e., the discretized singular curl-curl operator
\mathbf{K}_{cc}	part of the curl-curl matrix in the conductive subdomain Ω_c
\mathbf{K}_{cn}	coupling matrix between the conductive and nonconductive subdomains
\mathbf{K}_{nn}	part of the curl-curl matrix in the nonconductive subdomains Ω_n and Ω_{sc}
$\mathbf{K}_{nn,reg}$	regularized matrix \mathbf{K}_{nn}
\mathbf{K}_{nn}^+	representation of a pseudo-inverse of \mathbf{K}_{nn}
\mathbf{K}_S	generalized Schur complement
l	index
λ_{max}	maximum eigenvalue
l_K	total number of singular values of \mathbf{X}_K
$l_{\hat{K}}$	number of largest singular values of \mathbf{X}_K
l_M	total number of singular values of \mathbf{X}_M
$l_{\hat{M}}$	number of largest singular values of \mathbf{X}_M
m	index
μ	permeability
\mathbf{M}	mass matrix of conductivities
\mathbf{M}_{cc}	conductivity matrix
n	number of previous time steps
N_{CG}	number of maximally acceptable conjugate gradient iterations
Ω	spatial computational domain
Ω_c	subdomain of conducting material
Ω_n	subdomain of nonconducting material
Ω_{sc}	subdomain of excitation sources
\mathbf{r}	right-hand side vector
Σ_K	diagonal matrix in the POD method when computing improved start vectors for evaluating \mathbf{K}_{nn}^+
σ_K	singular value of \mathbf{X}_K
Σ_M	diagonal matrix in the POD method when computing improved start vectors for inverting \mathbf{M}_{cc}
σ_M	singular value of \mathbf{X}_M
t	time

τ	time constant
Δt	time step size
Δt_{CFL}	maximum stable time step size
Δt_{exp}	explicit time step size
Δt_{imp}	implicit time step size
tol	tolerance
\mathbf{U}_K	matrix in the POD method when computing improved start vectors for evaluating \mathbf{K}_{nn}^+
$\mathbf{U}_{\hat{K},\text{POD}}$	operator in the POD method when computing improved start vectors for evaluating \mathbf{K}_{nn}^+
\mathbf{U}_M	matrix in the POD method when computing improved start vectors for inverting \mathbf{M}_{cc}
$\mathbf{U}_{\hat{M},\text{POD}}$	operator in the POD method when computing improved start vectors for inverting \mathbf{M}_{cc}
\mathbf{V}_K	operator used for determining improved start vectors when evaluating \mathbf{K}_{nn}^+
\mathbf{v}_K	column vector of the operator \mathbf{V}_K
$\mathbf{v}_{K0,\text{SPE}}$	improved start vector when evaluating \mathbf{K}_{nn}^+ , computed using (C)SPE
\mathbf{V}_M	operator used for determining improved start vectors when inverting \mathbf{M}_{cc}
\mathbf{v}_M	column vector of the operator \mathbf{V}_M
$\mathbf{v}_{M0,\text{SPE}}$	improved start vector when inverting \mathbf{M}_{cc} , computed using (C)SPE
\mathbf{V}_{PK}	matrix in the POD method when computing improved start vectors for evaluating \mathbf{K}_{nn}^+
\mathbf{V}_{PM}	matrix in the POD method when computing improved start vectors for inverting \mathbf{M}_{cc}
x	position in space
\mathbf{x}	solution vector
$\mathbf{x}_{0,\text{POD}}$	improved start vector when inverting \mathbf{M}_{cc} , computed using POD
\mathbf{X}_K	snapshot matrix in the POD method when computing improved start vectors for evaluating \mathbf{K}_{nn}^+
\mathbf{X}_M	snapshot matrix in the POD method when computing improved start vectors for inverting \mathbf{M}_{cc}

\mathbf{x}_S	spatial distribution of the transient source current density i_S
\mathbf{z}_K	vector used in the (C)SPE method when computing improved start vectors for evaluating \mathbf{K}_{nn}^+
\mathbf{z}_M	vector used in the (C)SPE method when computing improved start vectors for inverting \mathbf{M}_{cc}

Symbols in Chapter 6

∇	Nabla operator
\mathbf{b}	right-hand side vector containing the transient Dirichlet boundary conditions
ε	permittivity
F	fine grid solution propagator
G	coarse grid solution propagator
h	height
I	I
j	index
k	iteration counter
κ	electric conductivity
\mathbf{K}	stiffness matrix
\mathbf{M}	mass matrix
N	number of available parallel processors
\mathbb{R}^n	n -dimensional space of real numbers
t	time
T_i	j -th point in time
Δt_c	time step size on the coarse time grid
Δt_f	time step size on the fine time grid
tol	tolerance
ϕ	electric scalar potential
\mathbf{x}	vector of time-dependent electric scalar potentials
\mathbf{x}_0	vector of initial values
$\bar{\mathbf{x}}_j$	solution vector on the coarse time grid at time T_i
$\tilde{\mathbf{x}}_j$	solution vector on the fine time grid at time T_i

\mathbf{X}_{j-1} vector of initial values for time integration on the coarse time grid
 \mathbf{X}_j^k vector of corrected initial values

1 Introduction

In the development process for electromagnetic devices, as e.g. electric machines, actuators, and transformers, computer-based numerical simulations of the time-dependent electromagnetic fields are a standard tool avoiding the necessity to build prototypes and perform measurements, which can both easily be time-consuming and expensive. The numerical simulations of these systems are usually based on the magnetoquasistatic approximation of Maxwell's equations.

The electroquasistatic approximation of Maxwell's equations is of relevance in the design of high voltage devices, e.g. surge arresters, AC/DC cable systems, or insulators. The robustness of high voltage devices can be improved by utilizing electric field stress grading microvaristor materials within the insulation. Due to a nonlinear field dependent conductivity, these materials become conductive if the electric field strength exceeds a switching field strength which prevents high voltage breakthroughs [34]. In the design process of such devices, transient nonlinear electroquasistatic field problems need to be solved numerically.

The required algorithms shall not only be robust and give sufficiently accurate results, but should also be computationally efficient. The goal is, that many simulations can be performed in an acceptable overall simulation time. Only if this criterion is matched, optimization algorithms that may require several thousands of simulations, as in case e.g. with genetic algorithms, may be employed to find an optimal design of the device to be developed. Furthermore, many simulations are required if uncertainty shall be quantified, e.g. if the effect of manufacturing tolerances of the dimension of an air gap in an electric machine on the torque shall be determined. Thus, the computational speed of the algorithms used for numerical simulations is directly related to low development costs and the ease with which an optimized design can be obtained.

The standard approach for the computation of the transient electromagnetic field problems follows the method of lines, i.e., the problem is firstly discretized in space and then secondly in time. A popular method for spatial discretization is the finite element method (FEM) due to its ability to use unstructured meshes and due to versatile ansatz functions suitable to various problems. Problems of interest frequently contain small structures, as e.g. the narrow air gap in an electric machine, or the spatial region around a triple point at the covered energized end of a long rod insulator, that need to be resolved using fine meshes resulting in large numbers of degrees of freedom (dofs). This also results in large algebraic systems of equations.

For integrating the spatially discretized field problem in time, various time integration methods exist. Amongst the most often used for integrating magneto- and or electroquasistatic field problems in time, are the implicit Euler method, the singly diagonal Runge-Kutta method (SDIRK3(2)) and Rosenbrock-Wanner methods. These all have in common that, being implicit methods, they are unconditionally stable which makes them applicable to the stiff ordinary differential equations of spatially discretized electroquasistatic field problems and to the infinitely stiff algebraic differential equations of discrete magnetoquasistatic field problems [45].

Commonly, ferromagnetic material is involved in magnetoquasistatic field problems of practical relevance, as e.g. in the design of an electric machine. This results in nonlinear systems of equations that need to be linearized as part of the solution process. The same problem holds for solving electroquasistatic field problems in the design of high-voltage insulators which might contain a nonlinear microvaristor field-grading material to prevent voltage breakthroughs. In order to solve a system of equations in an implicit time step, the system needs to be linearized. Linearization methods usually are iterative schemes, such as e.g. the Newton-Raphson method, which stop if changes in the sequence of approximative solutions are below a certain tolerance. In each implicit time step, several Newton-Raphson iterations might be necessary. In each Newton-Raphson iteration, the Jacobian matrix and the stiffness matrix must be updated. Thus, these iterative, implicit, unconditionally stable time integration schemes applied to large algebraic systems of equations, may result in a computational effort, that often requires long simulation times of up to several days to weeks for just one (usually three-dimensional, full model) configuration.

All of this contributes to long simulation times and might hinder the use in multi-query

scenarios, as e.g. multiparameter optimization or uncertainty quantification.

A significant acceleration of computations by further enhancement of clock speed of central processing units (CPUs) cannot be expected anymore, as exceeding power consumption make high clock speed inefficient and economically unattractive.

The aim of the work presented in this thesis is to establish algorithms with the potential to speed-up the time required for the simulation of magnetoquasistatic field problems and electroquasistatic field problems on contemporary and future hardware.

As the speed of a single processor is limited, it naturally follows that employing many processors respectively performing different tasks in parallel is a viable approach.

At this, the main idea underlying the work presented in this thesis, is to accelerate the computations by introducing algorithms that exploit the parallel-computing capabilities of general purpose graphics processing units (GPGPUs). The procedure described above, in which many large algebraic systems of equations may need to be linearized by an iterative method and are then commonly solved by an iterative solver, is not well suited for efficient parallelization. A key point to favor GPGPU-parallelization is the use of an explicit time integration method.

Explicit time integration methods favor parallelization by avoiding linearization, as nonlinearities commonly only occur in expressions on the right-hand side. Ideally, the computations required in an explicit time integration scheme are only matrix-vector multiplications which are well parallelizable on GPGPUs. An additional benefit of using explicit time integration methods is improved accuracy, due to the use of exact solvers and the strong oversampling caused by the commonly small time step sizes required for numerical stability. In semi-explicit time integration schemes linear algebraic systems of equations are solved, thus avoiding the use of linearization methods.

Another promising approach to reducing simulation time by exploiting the available many-core hardware, is integration in time using the parallel-in-time algorithm, also termed Parareal algorithm. This approach is investigated for electroquasistatic field problems in this thesis. In the Parareal algorithm, the time axis is discretized by a coarse and by a fine grid. In a first phase, the spatially discretized problem is integrated in time using the coarse time grid, i.e., using a large time step size, by an unconditionally stable implicit time integration scheme. In a second phase, the problem is

integrated in time using an accurate explicit time integration scheme, i.e., using a small time-step size, between the grid points of the coarse time grid. Since the time integration in each time interval of the coarse time grid can be performed independently from the other intervals, the computations can be carried-out in parallel on different hardware units, e.g. CPU cores, GPGPU cores, or even server nodes. In a third phase, the solution on the coarse time grid points is improved based on the results from the solutions on the fine time grid points and a difference between the old and the new solution on the coarse time grid points is evaluated. The first, second, and third phase of the Parareal algorithm are repeated iteratively until convergence of the solution is reached, i.e., the difference evaluated in the third phase reaches values lower than a chosen tolerance.

1.1 Previous and Related Works

Early work by Yioultsis et al. investigate the use of explicit time integration methods for the numerical computation of magnetoquasistatic field problems, starting from the finite difference time domain (FDTD) method in [117, 116]. The FDTD scheme probably is the most popular method for numerically computing electromagnetic field problems based on the wave-equation, i.e., for computing radiation and scattering problems [115]. For these problems, based on the full set of Maxwell's equations, the interdependence of electric and magnetic fields is very well represented by the leap-frog scheme. In case of the electroquasistatic approximation, the magnetic induction is neglected, whilst the displacement current is neglected in the magnetoquasistatic approximation. Hence, applying the FDTD method's leap frog scheme is less straight-forward for quasistatic field problems. An additional disadvantage is the frequent presence of nonlinear materials in quasistatic field problems, as e.g., ferromagnetic material. Approaches such as Scaled Lightspeed FDTD, which keep the displacement current and scale the problem to a reduced speed of light still have a very small stable time step size.

Yioultsis et al. derive a numerical computation method in [117] based on the FDTD method, termed "General FDTD", usable in magnetoquasistatic field problems. Here, a modified magnetic vector potential formulation is used in the conducting regions

of the problems, where eddy currents are excited, while a magnetic scalar potential formulation is considered in the nonconducting regions of the problem. The problem is discretized in space using finite differences on a structured grid. As the magnetoquasistatic field problem in the conducting regions is a diffusion problem, the explicit DuFort-Frankel algorithm is chosen for time integration in the conducting regions, which is suitable for diffusion problems. In the nonconducting regions, the magnetoquasistatic problem is reduced to a magnetostatic Poisson problem. Here, two approaches are proposed by Youltsis et al. The first was to compute the field in the nonconducting regions in the outside domains of the problem using the boundary element method (BEM) [117]. The BEM, however, requires inverting a large dense matrix. Although the inversion is only performed once, storing the inverse requires a lot of computer memory. Alternatively, matrix sparsification techniques, such as adaptive cross approximation (ACA) [5], could be used to make BEM formulations accessible to iterative solution techniques. As an alternative approach to the BEM, terminating the computational domain using a perfectly matched layer (PML) is proposed in [116]. The PML has the additional advantage of efficiently truncating the mesh at the computational domain's boundaries. The main advantage of allowing the use of an explicit time integration scheme, comes at the cost of using two different methods and formulations for spatially discretizing the conducting and nonconducting regions, which results in additional algebraic equation systems at the boundaries between conducting and nonconducting regions for matching results separately obtained in both regions.

Conducting and nonconducting regions of eddy current problems are also treated separately in the work by Außerhofer et al. presented in [2]. As in [117, 116], a magnetic vector potential formulation is used. The conducting regions are spatially discretized using the discontinuous Galerkin finite element method (DF-FEM). The uncoupled finite elements yield a block diagonal mass matrix and enable explicit time stepping. The stiffness of the system of ordinary differential equations describing the magnetic vector potential in the conducting regions requires a small stable time step size. The nonconducting regions are discretized in space by FEM with continuous ansatz functions. In nonconducting regions, no eddy currents exist and magnetic vector potential and magnetic field do not have a time derivative. Therefore, no actual time integration is performed for the field in these regions, but it is updated regularly to match the field in the conducting regions at the boundary conditions between both regions. At this,

an equation system containing the ill-conditioned curl-curl matrix needs to be solved. The proposed method is demonstrated on a two-dimensional numerical example [2].

A similar approach is proposed in [99, 100, 97]. Based on an $H - \Phi$ -formulation DG-FEM, where H is the magnetic field strength and Φ is the magnetic scalar potential, is used for discretizing conducting regions in space. For spatially discretizing the magnetic scalar potential Φ in nonconducting regions, FEM based on continuous nodal elements is applied. This eliminates the ill-conditioned curl-curl matrix in the nonconducting regions and an equation system based on well-conditioned matrices needs to be solved when the field in the nonconducting regions is updated. As in [2, 3], an explicit method is used for time integration in the conducting regions and the field in the nonconducting regions is updated only after several explicit time steps. The method is demonstrated in [99] on a three-dimensional example with constant magnetic permeability, where the maximum stable time step size for the explicit time integration scheme is in the micro-second range. The application of this method to multiply connected domains is presented in [98], where a multivalued magnetic scalar potential, as in [110], is used for representing the rotational part of the magnetic field in nonconducting, multiply-connected regions. A three-dimensional FEM model of a solid conductor bend with an elliptical hole is used for numerical validation.

So far, all previous works presented are based on the idea to treat conducting and nonconducting regions separately by discretizing them in space using different methods. The work presented in [14, 92] is based on a different approach and is the fundament on which this thesis is based on. A magnetic vector potential formulation of the magnetoquasistatic field approximation is used. The entire computational domain is discretized in space using the same method, e.g. FEM or the finite integration technique (FIT) [112, 113]. By separating the unknowns depending on their allocation in conducting or nonconducting material and application of a Schur complement, the differential algebraic equation system is transformed into an ordinary differential equation system. This stiff ordinary differential equation system can be integrated in time using an explicit time integration scheme. The approach will be explained in detail in chapter 5.

With respect to accelerating the simulation time for solving electroquasistatic field problems, several methods are presented in [78], covering both computational aspects,

as e.g. applying GPU-accelerated FEM assembly or GPU-accelerated multigrid preconditioning techniques, as well as mathematical approaches, as e.g. explicit time integration methods, especially Runge-Kutta-Chebyshev methods [103, 79, 78]. Persistent developments in computer hardware towards massive many-core systems promise significant computational speed-up for well parallelizable algorithms, e.g. the parareal algorithm for parallel-in-time integration proposed in [62, 41, 40, 39]. Here, time integration is performed adopting a multigrid approach: In a first step, time integration is carried out serially on a coarse time grid to obtain initial values for time integration on a fine grid, where each interval between two time points on the coarse grid is given to an individual processor. In a second step, all processors perform a time integration on the fine time grid in parallel. Integration on the coarse and fine time grid is executed iteratively until convergence.

The work in this thesis follows the approach presented in [94] where the Parareal algorithm is applied to a transient nonlinear magnetoquasistatic field problem using the implicit Euler method for time integration on both the coarse and the fine time grid. In this work, the application of the explicit Euler scheme on the fine time grid is additionally investigated.

1.2 Overview

This thesis's main focus is on formulating and accelerating the explicit time integration of three-dimensional nonlinear transient magnetoquasistatic field problems based on a magnetic vector potential formulation that are transformed into ordinary systems of equations, as originally suggested in [14, 92].

Additionally, the applicability of the Parareal algorithm to nonlinear two-dimensional electroquasistatic field problems is presented.

The structure of this thesis is outlined in the following.

Chapter 2 is an introduction to the partial differential equations considered and explains the derivation of the magneto- and electroquasistatic approximation starting from Maxwell's equations.

The discretization in space and integration in time of both partial differential equations derived in chapter 2 is subject to chapter 3. Here, the application of the finite element method for the spatial discretization and of the explicit and implicit Euler method for time integration are briefly explained.

Two different solvers for solving large systems of algebraic equations are used within the work of this thesis: The preconditioned conjugate gradient method and the LU decomposition. Both are highlighted in chapter 4.

Accelerating techniques for the (semi-) explicit time integration of magnetoquasistatic field problems are presented in chapter 5. The approach for transforming the differential algebraic equation (DAE) system of the magnetoquasistatic field problem based on a magnetic vector potential formulation into an ordinary differential system of equations as suggested in [14, 92] is explained in detail in subchapter 5.1. The transformation requires a pseudo-inversion of the singular curl-curl matrix in nonconducting regions. The singular matrix can be regularized as shown in [14, 92] using a grad-div gauging technique. Alternatively, a pseudo-inverse can be used by applying the preconditioned conjugate gradient method. This approach was first presented in [25]. It is used as basis for further investigations on acceleration performed within the framework of this thesis and explained in chapter 5.

The computation of improved start vectors for the preconditioned conjugate gradient method as a first measure to accelerate computations is described in subchapter 5.2. The algebraic equation system to be solved for evaluating a pseudo-inverse, as well as the equation system to be solved for inverting the mass matrix, resembles a multiple right-hand side problem. The solution process can be accelerated by applying multiple right-hand side techniques, such as the subspace projection extrapolation method [17]. The aim here, is to compute improved start vectors for the preconditioned conjugate gradient method based on previous solutions. This can be modified to the cascaded subspace projection extrapolation method as published e.g. in [30, 31, 32]. Alternatively, the proper orthogonal decomposition can be used for this purpose, as presented in [29].

Additionally, it is possible to bypass updating the reluctivity matrix in some time steps due to the small stable time step size, which yields an additional speed-up. This idea was presented e.g. in [26, 24]. It is presented in detail in subchapter 5.3.

Investigations show that a computationally cheap regularization of the singular curl-curl matrix in nonconducting media is possible by adding a small offset-value to its main diagonal elements, i.e., considering a perturbed system formulation. As a result, the matrix can be inverted using a direct method, as e.g. the LU-decomposition, which is well parallelizable on GPUs. This approach was presented in [27] and is subject to subchapter 5.4.

Chapter 6 presents time integration schemes for electroquasistatic field problems using the Parareal algorithm as published in [94]. The approach is explained and numerically investigated using a two-dimensional finite element model of an IEC 60099-4 surge arrester [55]. Various combinations of the implicit and explicit Euler method for time integration on the fine and on the coarse time grid are investigated within this framework.

Finally, conclusions and an outlook on possible future research topics are presented in chapter 7.

2 Mathematical Description of Electromagnetic Field Problems

This chapter describes how the algebraic systems of equations resulting from the spatial discretization of the problem considered are obtained. Starting from Maxwell's equations, the approximation applicable to magnetoquasistatic field problems is derived first. Its spatial discretization by the finite element method is described, in succession of explaining the finite element method. Further relevant information on treating a spatially discretized computational model with respect to the application of boundary conditions and the partitioning into different material sections are given. Integration in time is subject to the next chapter.

2.1 Maxwell's Equations

Electromagnetic field problems are described by the four partial differential equations

$$\nabla \times \vec{E}(\vec{r}, t) = -\frac{\partial}{\partial t} \vec{B}(\vec{r}, t), \quad (2.1)$$

$$\nabla \times \vec{H}(\vec{r}, t) = \frac{\partial}{\partial t} \vec{D}(\vec{r}, t) + \vec{J}(\vec{r}, t), \quad (2.2)$$

$$\nabla \cdot \vec{D} = \varrho(\vec{r}, t), \quad (2.3)$$

$$\nabla \cdot \vec{B} = 0, \quad (2.4)$$

at each location $\vec{r} \in \Omega \subseteq E^3$ and time $t \in (t_0, t_{\text{end}}]$, known as Maxwell's equations [68], stated here using the notation introduced by Heaviside in [46, 47]. Here, \vec{E} is the electric field strength, \vec{B} is the magnetic flux density, \vec{H} is the magnetic field strength, \vec{D} is the electric flux density, \vec{J} is the current density, and ϱ is the electric charge

density. Additionally, initial and boundary conditions are required for solving field problems described by equations (2.1) - (2.4).

Further relations between Maxwell's equations are provided by the constitutive material relations

$$\vec{D} = \varepsilon_0 \varepsilon_r \vec{E} + \vec{P}_E, \quad (2.5)$$

$$\vec{B} = \mu_0 \mu_r \vec{H} + \vec{P}_M, \quad (2.6)$$

$$\vec{J} = \kappa \vec{E} + \vec{J}_S, \quad (2.7)$$

where ε_0 is the electric constant, ε_r is the relative permittivity, \vec{P}_E is the electric polarization vector, μ_0 is the magnetic constant, μ_r is the relative permeability, \vec{P}_M is the magnetic polarization vector, κ is the electric conductivity and \vec{J}_S is the impressed current density. Alternatively, equation (2.6) can also be expressed as

$$\vec{H} = \nu \vec{B} - \nu \vec{P}_M, \quad (2.8)$$

by using the magnetic reluctivity $\nu = (\mu_0 \mu_r)^{-1}$.

The material properties ε_r , μ_r , κ , and ν can also depend on space, i.e., $\varepsilon_r = \varepsilon_r(\vec{r})$, $\mu_r = \mu_r(\vec{r})$, $\kappa = \kappa(\vec{r})$, and $\nu = \nu(\vec{r})$. Furthermore, the material properties of some materials are also field dependent. This field dependence can be linear or nonlinear. The permeability of steel, e.g., nonlinearly depends on the magnetic field. In high voltage devices, nonlinear field grading material, as e.g. ZnO microvaristors, can be used to avoid high voltage breakdowns. Here, the electric conductivity nonlinearly depends on the electric field strength. A plot of a nonlinear ferromagnetic B - H -curve of steel is shown in Figure 5.9. A conductivity curve of a nonlinear field grading material is depicted in Figure 6.10. The permeability, respectively the electric conductivity, needs to be evaluated locally depending on the corresponding field strength in numerical simulations, where these nonlinear curves are usually interpolated by cubic splines [59].

Maxwell's equations (2.1) - (2.4) can also be expressed using the magnetic vector potential \vec{A} and the electric scalar potential ϕ . The magnetic flux density \vec{B} and the electric field strength \vec{E} can be described using the electrodynamic potentials (\vec{A}, ϕ)

by

$$\vec{B} = \nabla \times \vec{A}, \quad (2.9)$$

$$\vec{E} = -\nabla\phi - \frac{\partial}{\partial t}\vec{A}. \quad (2.10)$$

If the magnetic vector potential \vec{A} is used, an additional gauging condition is required, e.g. the frequently chosen Coulomb's gauge $\nabla \cdot \vec{A} = 0$, since the magnetic vector potential \vec{A} is defined by the magnetic flux density \vec{B} only up to a gradient field.

2.2 The Magnetoquasistatic and Electroquasistatic Approximation

Solving the complete set of Maxwell's equations can often be avoided in technical applications by using approximations valid in the considered frequency range. Quasistatic approximations may be used, if the dimensions of the considered object are small compared to the wavelength [23]. Here, the magnetoquasistatic (MQS) approximation is presented first, which is valid if additionally to the above stated dimension criterion, $W_M \gg W_E$, $\forall(\vec{r}, t) \in \Omega \times [t_0, t_{\text{end}}]$ [23], where W_M is the magnetic energy, W_E is the electric energy, Ω is the spatial computational domain, t_0 is the start point in time, and t_{end} is the end point in time of the time interval considered. If these prerequisites hold, the displacement current $\partial\vec{D}/\partial t$ can be neglected with respect to equation (2.2). By using equation (2.9) in equation (2.2) the MQS field formulation

$$\kappa \frac{\partial}{\partial t} \vec{A} + \nabla \times \left[\nu (\nabla \times \vec{A}) \nabla \times \vec{A} \right] = \vec{J}_S, \quad \forall(\vec{r}, t) \in \Omega \times (t_0, t_{\text{end}}], \quad (2.11)$$

where \vec{J}_S is the source current density, is obtained. Equation (2.11) is a parabolic partial differential equation, if $\kappa > 0$ and a semi-elliptic partial differential equation otherwise. Solving equation (2.11) for the magnetic vector potential \vec{A} requires imposing a temporal initial value and spatial boundary conditions. Within the work of this thesis, the initial value $\vec{A}(\vec{r}, t = 0) = \vec{0}$ is imposed. Alternatively, e.g. a solution for the magnetostatic approximation can be used.

The electroquasistatic approximation is applicable in a similar scenario, i.e., the dimensions of the considered object are small compared to the wavelength and if $W_E \gg$

$W_M, \forall(\vec{r}, t) \in \Omega \times [t_0, t_{\text{end}}]$ [23]. In this case, the magnetic induction $\partial\vec{B}/\partial t$ is considered to be negligible in equation (2.1) and the magnetic vector potential \vec{A} can be omitted in equation (2.10). Applying the divergence to equation (2.2) results in the continuity equation

$$\nabla \cdot \left(\frac{\partial}{\partial t} \vec{D}(\vec{r}, t) \right) + \nabla \cdot \vec{J}(\vec{r}, t) = 0, \quad \forall(\vec{r}, t) \in \Omega \times (t_0, t_{\text{end}}]. \quad (2.12)$$

Replacing the electric flux density \vec{D} with $\vec{D} = \varepsilon(-\nabla\phi)$ and the electric current density \vec{J} with $\vec{J} = \kappa(-\nabla\phi)$ in equation (2.12) yields the electroquasistatic approximation

$$\nabla \cdot (\kappa\nabla\phi) + \nabla \cdot \left(\varepsilon\nabla\frac{\partial\phi}{\partial t} \right) = 0, \quad \forall(\vec{r}, t) \in \Omega \times (t_0, t_{\text{end}}], \quad (2.13)$$

based on the time-dependent electric scalar potential ϕ . Both equations (2.13) and (2.11) are defined on the domain $\Omega \times I$, where Ω is the spatial computational domain and $I := [t_0, t_{\text{end}}]$ is the time interval considered.

3 Discretization of Partial Differential Equations in Space and Time

The discretization of both partial differential equations (2.13) and (2.11) follows the method of lines [90]. Here, the partial differential equation to be solved is first discretized in one variable, i.e., the space variable, while the time variable remains continuous [87]. In this thesis, the PDEs (2.13) and (2.11) are first spatially discretized using the Finite Element Method (FEM). Integration in time is then realized either by the implicit or by the explicit Euler method or Parareal variants thereof.

In this chapter, the application of the FEM to (2.13) and (2.11) is presented first in subchapter 3.1. In the succeeding subchapter 3.2, the implicit and the explicit Euler method are generally introduced and the application of the implicit Euler scheme to the spatially discretized magneto- and electroquasistatic PDEs and of the explicit Euler method to the electroquasistatic PDE is explained. The application of the explicit Euler method to magnetoquasistatic field problems is subject to chapter 5.

3.1 The Finite Element Method

The Finite Element Method (FEM) is a numerical method for spatially discretizing partial differential equations in many engineering applications, as e.g. (quasistatic) electromagnetics, structural analysis or fluid dynamics. The FEM is an established technique in literature, see e.g. [4, 96, 120, 52, 69, 61, 56]. Details on the computational implementation, such as e.g. the use of a reference element for numerical integration, are e.g. explained in the corresponding chapter on FEM in [85].

In this chapter, the application of the FEM to the partial differential equations (2.11) and (2.13) is briefly presented.

3.1.1 FEM applied to the PDE of Magnetoquasistatic Field Problems

The application of the FEM to magnetoquasistatic field problems presented in the following, follows the explanations in [114]. A vector test function \vec{v} from the Sobolev space $H_0(\text{curl}, \Omega)$ is used in the following, where

$$H(\text{curl}, \Omega) := \{\vec{v} \in L^2(\Omega) : \nabla \times \vec{v} \in L^2(\Omega)\}, \quad (3.1)$$

$$H_0(\text{curl}, \Omega) := \{\vec{v} \in H(\text{curl}, \Omega) : \vec{v} \times \vec{n} = 0 \text{ on } \Gamma = \partial\Omega\}. \quad (3.2)$$

Here, $L^2(\Omega)$ is the space of square-integrable functions and Γ denotes the boundary of the computational domain Ω and \vec{n} is the outward pointing normal vector of the boundary Γ .

Multiplying the partial differential equation (2.11) with a test function $\vec{v} \in H_0(\text{curl}, \Omega)$ and integrating over the domain Ω yields

$$\int_{\Omega} \kappa \frac{\partial \vec{A}}{\partial t} \cdot \vec{v} \, d\Omega + \int_{\Omega} (\nabla \times [\nu(\vec{B}) \nabla \times \vec{A}]) \cdot \vec{v} \, d\Omega = \int_{\Omega} \vec{J}_S \cdot \vec{v} \, d\Omega. \quad (3.3)$$

The notation $\nu(\vec{B})$ represents the dependence of the magnetic reluctivity ν on the magnetic flux density \vec{B} .

Applying Green's theorem for vectors to equation (3.3) results in

$$\int_{\Omega} \kappa \frac{\partial \vec{A}}{\partial t} \cdot \vec{v} \, d\Omega + \int_{\Omega} \nu(\vec{B}) (\nabla \times \vec{A}) \cdot (\nabla \times \vec{v}) \, d\Omega = \int_{\Omega} \vec{J}_S \cdot \vec{v} \, d\Omega. \quad (3.4)$$

Now, the solution vector $\vec{A} \in H_0(\text{curl}, \Omega)$ is to be computed, such that

$$a(\vec{A}, \vec{v}) + b(\vec{A}, \vec{v}) = r(\vec{v}) \quad \forall \vec{v} \in H_0(\text{curl}, \Omega), \quad (3.5)$$

holds, where

$$a(\vec{A}, \vec{v}) = \int_{\Omega} \nu(\vec{B}) (\nabla \times \vec{A}) \cdot (\nabla \times \vec{v}) \, d\Omega, \quad (3.6)$$

$$b(\vec{A}, \vec{v}) = \int_{\Omega} \kappa \frac{\partial \vec{A}}{\partial t} \cdot \vec{v} \, d\Omega, \quad (3.7)$$

and

$$r(\vec{v}) = \int_{\Omega} \vec{J}_S \cdot \vec{v} \, d\Omega. \quad (3.8)$$

In order to achieve this, equation (3.4) is discretized e.g. using lowest, i.e., first order edge elements basis functions

$$\vec{w}_{ij}^{(1)} := N_i \nabla N_j - N_j \nabla N_i, \quad (3.9)$$

which are assigned to the edges of the finite element mesh [114]. The nodal basis functions N_i and N_j are assigned to the nodes i and j connected by an edge E_{ij} . The line integral of the edge element basis function $\vec{w}_{ij}^{(1)}$ is only nonzero along the edge E_{ij} and zero along all other edges in the mesh connecting different nodes k and l . That is

$$\int_{E_{ij}} \vec{t}_{ij} \cdot \vec{w}_{ij}^{(1)} \, ds = 1, \quad (3.10)$$

$$\int_{E_{kl}} \vec{t}_{kl} \cdot \vec{w}_{ij}^{(1)} \, ds = 0 \text{ for } E_{kl} \neq E_{ij}, \quad (3.11)$$

where the unit vector \vec{t}_{ij} points along the edge E_{ij} from node i to node j and the unit vector \vec{t}_{kl} points along the edge E_{kl} from node k to node l .

Using these edge elements, \vec{A} can be expressed by the linear combination

$$\vec{A} \approx \vec{A}_n = \sum_{i=1}^{n_e} a_i \vec{w}_{ei}^{(1)}, \quad (3.12)$$

where n_e is the total number of edges in the mesh and $\vec{w}_{ei}^{(1)}$ is the basis function assigned to the i -th edge in the mesh. By using the basis functions $\vec{w}_{ej}^{(1)}$ also as test functions \vec{v} , equation (3.5) is reformulated as a system of nonlinear differential algebraic equations

$$\mathbf{M} \frac{d}{dt} \mathbf{x} + \mathbf{K}(\mathbf{x}) \mathbf{x} = \mathbf{r}, \quad (3.13)$$

where the entries in the i -th row and in the j -th column of the matrices \mathbf{M} and \mathbf{K} are as follows:

$$\mathbf{M}^{(i,j)} = b(\vec{w}_{ei}^{(1)}, \vec{w}_{ej}^{(1)}) = \int_{\Omega} \kappa \vec{w}_{ei}^{(1)} \cdot \vec{w}_{ej}^{(1)} \, d\Omega, \quad (3.14)$$

$$\mathbf{K}^{(i,j)} = a(\vec{w}_{ei}^{(1)}, \vec{w}_{ej}^{(1)}) = \int_{\Omega} (\nabla \times \vec{w}_{ei}^{(1)})^{\top} \cdot \nu(\vec{B}) (\nabla \times \vec{w}_{ej}^{(1)}) \, d\Omega. \quad (3.15)$$

The j -th entry of the vector \mathbf{r} is computed as

$$\mathbf{r}^{(j)} = r(\vec{w}_{ej}^{(1)}) = \int_{\Omega} \vec{J}_S \cdot \vec{w}_{ej}^{(1)} \, d\Omega. \quad (3.16)$$

The vector \mathbf{x} contains the degrees of freedom, i.e., the unknowns a_i , for which the system of equations (3.13) is solved, $\mathbf{x}(t) = \{a_i(t)\}_{i=1, \dots, n_e}$.

The integrals in equations (3.14) - (3.16) are usually evaluated numerically using e.g. Gauss-Legendre quadrature [101].

3.1.2 FEM Applied to the PDE of Electroquasistatic Field Problems

For the EQS problem (2.13), a scalar test function v from the Sobolev space $H^1(\Omega)$ is used, where

$$v \in H^1(\Omega) := \{v \in L^2(\Omega), \nabla v \in L^2(\Omega)\} \quad (3.17)$$

holds. Here, $L^2(\Omega)$ is the space of square-integrable functions.

In a first step, the PDE (2.13) is multiplied with the test function v and their product is integrated over the whole computational domain Ω , which results in

$$\int_{\Omega} v \cdot (\nabla \cdot (\kappa(\phi) \nabla \phi)) \, d\Omega + \int_{\Omega} v \cdot \left(\nabla \cdot \left(\varepsilon \nabla \frac{\partial \phi}{\partial t} \right) \right) \, d\Omega = 0. \quad (3.18)$$

Applying Green's first theorem to equation (3.18) yields

$$\begin{aligned} & \int_{\Omega} \nabla v \cdot (\kappa(\phi) \nabla \phi) \, d\Omega + \int_{\Omega} \nabla v \cdot \left(\varepsilon \nabla \frac{\partial \phi}{\partial t} \right) \, d\Omega \\ &= \int_{\partial \Omega} \left(\kappa(\phi) \frac{\partial \phi}{\partial \vec{n}} \cdot v + \varepsilon \frac{\partial}{\partial \vec{n}} \left(\frac{\partial \phi}{\partial t} \right) \cdot v \right) \, dS. \end{aligned} \quad (3.19)$$

Here, \vec{n} is the outward pointing normal vector of the boundary Γ of the computational domain Ω . Usually, and also in case of the work presented in this thesis, homogeneous Neuman boundary conditions are applied to the boundary Γ of Ω , i.e., $\frac{\partial \phi}{\partial \vec{n}}|_{\Gamma} = 0$. Therefore, the right-hand side of equation (3.19) equals zero and equation (3.19) becomes

$$\int_{\Omega} \nabla v \cdot (\kappa(\phi) \nabla \phi) \, d\Omega + \int_{\Omega} \nabla v \cdot \left(\varepsilon \nabla \frac{\partial \phi}{\partial t} \right) \, d\Omega = 0. \quad (3.20)$$

Equation (3.20) constitutes the variational problem of finding ϕ , so that equation (3.20) holds for all $v \in H^1(\Omega)$. Equation (3.20) is discretized by approximating $\phi = \phi(\vec{r}, t)$ and $v = v(\vec{r})$ using nodal basis functions

$$N_j(\vec{r}_i) = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}, \quad i, j = 1, \dots, n_n, \quad (3.21)$$

where n_n is the total number of nodes in the finite element mesh at the positions \vec{r}_i . The j -th nodal basis function $N_j(\vec{r}_i)$ is assigned to the i -th node of the mesh at the spatial position \vec{r}_i . Thus, the electric vector potential $\phi = \phi(\vec{r}, t)$ is approximated by

$$\phi(\vec{r}, t) \approx \phi_n = \sum_{j=1}^{n_n} u_j(t) N_j(\vec{r}), \quad (3.22)$$

where $u_j(t)$ are the unknowns, also known as degrees of freedom for which the discretized variational problem

$$\sum_{j=1}^{n_n} \left(u_j \int_{\Omega} (\nabla N_i) \cdot \kappa(u_j) \cdot (\nabla N_j) \, d\Omega + \frac{\partial}{\partial t} u_j \int_{\Omega} (\nabla N_i) \cdot \varepsilon \cdot (\nabla N_j) \, d\Omega \right) = 0 \quad (3.23)$$

needs to be solved, thus equation (3.23) is the discretized version of equation (3.20).

Equation (3.23) constitutes a nonlinear system of ordinary equations described by

$$\mathbf{M} \frac{d}{dt} \mathbf{x} + \mathbf{K}(\mathbf{x}) \mathbf{x} = 0. \quad (3.24)$$

Here, the matrix entries of the i -th row and j -th column of the matrices \mathbf{M} and $\mathbf{K}(\mathbf{x})$ are computed by

$$\mathbf{M}^{(i,j)} = \int_{\Omega} (\nabla N_i) \cdot \varepsilon (\nabla N_j) \, d\Omega \quad (3.25)$$

and by

$$\mathbf{K}^{(i,j)} = \int_{\Omega} (\nabla N_i) \cdot \kappa(u_j) (\nabla N_j) \, d\Omega. \quad (3.26)$$

For practical computations, the matrix entries resulting from nodes to which eventually transient Dirichlet boundary conditions are applied, are extracted from the matrices \mathbf{M} and \mathbf{K} and transferred to the right-hand side of equation (3.24), which becomes

$$\mathbf{M} \frac{d}{dt} \mathbf{x} + \mathbf{K}(\mathbf{x}) \mathbf{x} = \mathbf{r}(t), \quad (3.27)$$

where $\mathbf{r}(t) = -\mathbf{M}_D \frac{d}{dt} \mathbf{x}_D - \mathbf{K}_D(\mathbf{x}_D) \mathbf{x}_D$. Here, the matrices \mathbf{M}_D and \mathbf{K}_D only have nonzero entries corresponding to the nodes on which Dirichlet boundary conditions are applied and the vector \mathbf{x}_D contains the Dirichlet boundary conditions.

3.2 Time Integration Methods

In this subchapter, the implicit and the explicit Euler method are briefly introduced. Their application to the spatially discretized magneto- and electroquasistatic PDEs is explained, except for the application of the explicit Euler method to magnetoquasistatic field problems. This is subject of chapter 5.

3.2.1 The Implicit Euler Method

The implicit Euler method is applied to an ordinary differential equation (ODE)

$$\frac{dy}{dt} = f(y, t), \quad y(t=0) = y_0, \quad (3.28)$$

where y_0 is the initial value, by evaluating

$$y_{i+1} = y_i + \Delta t f(y_{i+1}, t_{i+1}). \quad (3.29)$$

Here i denotes the i -th time step at time t_i , y_i is the discretization of the continuous function $y(t = t_i)$, Δt is the constant time step size between two discrete points in time, i.e., $\Delta t = t_{i+1} - t_i$, and y_{i+1} is the discretization of the continuous function

$y(t = t_{i+1} = t_i + \Delta t)$. The implicit Euler method's stability region comprises the complex plane except for a circular disk with radius 1 centered at 1, which is excluded from the stability region, as shown in Figure 3.1. The implicit Euler method is L-stable and stiffly accurate [45, 10]. It is thus well suited for the time integration of stiff and even infinitely stiff problems, such as the system of differential algebraic equations (3.13) encountered in the computation of magnetoquasistatic field problems. With respect to stability, an arbitrarily selected time step size can be used in the implicit Euler scheme.

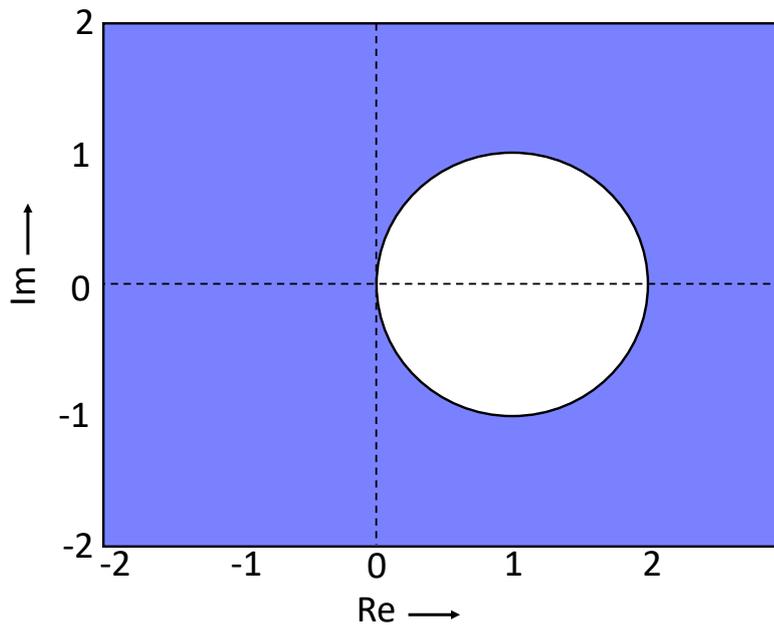


Figure 3.1: Stability region of the implicit Euler method in the z -plane, as e.g. in [45]. The region of stability is colored in blue.

In the following, the application of the implicit Euler method to the system of differential algebraic equations (3.13) describing spatially discretized magnetoquasistatic field problems and to the system of ordinary differential equations (3.27) describing electroquasistatic field problems is explained, following [13]. Since both equations (3.13) and (3.27) are of the same form

$$\mathbf{M} \frac{d}{dt} \mathbf{x} + \mathbf{K}(\mathbf{x}) \mathbf{x} = \mathbf{r}, \quad (3.30)$$

the application of the implicit Euler method is explained for equation (3.30) representing either equation (3.13) or (3.27).

It is assumed that the solution \mathbf{x}_i at the i -th time step at time t_i is already known. This is represented by the notation $\mathbf{x}_i \approx \mathbf{x}(t_i)$. For evaluating the solution \mathbf{x}_{i+1} , i.e., the $(i+1)$ -th time step solution, at time $t_{i+1} = t_i + \Delta t$, the nonlinear algebraic system of equations

$$\left(\frac{1}{\Delta t} \mathbf{M} + \mathbf{K}(\mathbf{x}_{i+1}) \right) \mathbf{x}_{i+1} = \frac{1}{\Delta t} \mathbf{M} \mathbf{x}_i + \mathbf{r}_{i+1} \quad (3.31)$$

must be solved for \mathbf{x}_{i+1} . Here, the Newton-Raphson method is used for linearization. In the $(l+1)$ -th Newton-Raphson iteration, the linear system of equations

$$\left(\frac{1}{\Delta t} \mathbf{M} + \mathbf{J}(\mathbf{x}_{i+1}^l) \right) \mathbf{x}_{i+1}^{l+1} = \frac{1}{\Delta t} \mathbf{M} \mathbf{x}_i^l + \mathbf{r}_{i+1} + \mathbf{J}(\mathbf{x}_{i+1}^l) \mathbf{x}_{i+1}^l - \mathbf{K}(\mathbf{x}_{i+1}^l) \mathbf{x}_{i+1}^l \quad (3.32)$$

is solved for \mathbf{x}_{i+1}^{l+1} , where the Jacobian matrix \mathbf{J} is computed by

$$\mathbf{J}(\mathbf{x}) = \frac{d}{d\mathbf{x}} (\mathbf{K}(\mathbf{x})\mathbf{x}). \quad (3.33)$$

3.2.2 The Explicit Euler Method

The explicit Euler method is applied to an ordinary differential equation (ODE)

$$\frac{dy}{dt} = f(y, t), \quad y(t=0) = y_0, \quad (3.34)$$

where y_0 is the initial value, by evaluating

$$y_{i+1} = y_i + \Delta t f(y_i, t_i), \quad (3.35)$$

where i denotes the i -th time step at time t_i , y_i is the discretization of the continuous function $y(t = t_i)$, Δt is the time step size between to discrete points in time, i.e., $\Delta t = t_{i+1} - t_i$, and y_{i+1} is the discretization of the continuous function $y(t = t_{i+1} = t_i + \Delta t)$.

The explicit Euler method's stability region is a circular disk with radius 1 centered at -1 in the complex plane, as shown in Figure 3.2 [45, 10]. It is stable only for time step sizes smaller than a maximum stable time step size, which is stated by the Courant-Friedrichs-Lewy criterion [19].

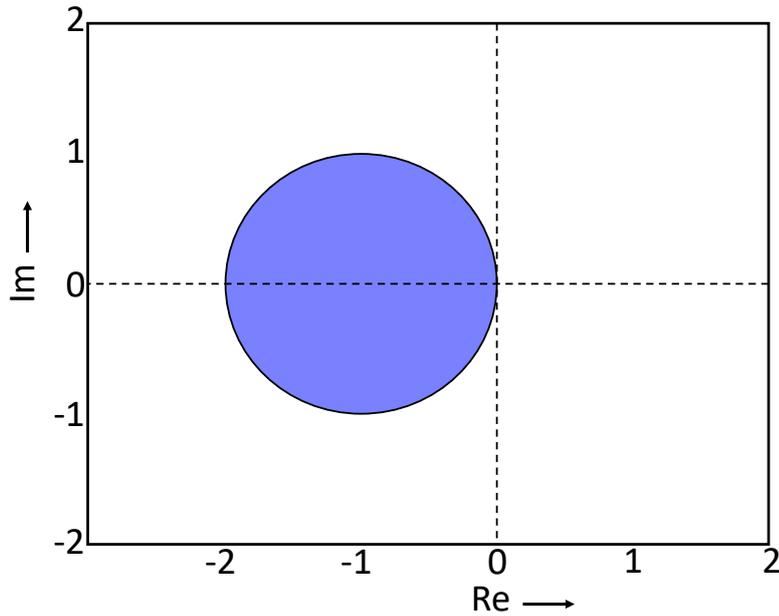


Figure 3.2: Stability region of the explicit Euler method in the z -plane, as e.g. in [45]. The region of stability is colored in blue.

Assuming non-singularity of the matrix \mathbf{M} and applying the explicit Euler method to the system of ordinary differential equations (3.27) describing electroquasistatic field problems, a solution at the $(i + 1)$ -th time step is simply obtained by evaluating

$$\mathbf{x}_{i+1} = \mathbf{x}_i + \Delta t \cdot \mathbf{M}^{-1} [\mathbf{r}_{i+1} - \mathbf{K}(\mathbf{x}_i) \mathbf{x}_i]. \quad (3.36)$$

From an engineering point of view, the ease of implementation is the explicit Euler method's main advantage. Yet, the small stability region reduces its applicability for real-world problems. Methods which overcome this disadvantage are eventually higher-order, stabilized explicit Runge-Kutta methods, as described e.g. in [45, 103, 111]. However, these are not implemented in this thesis.

The application of the explicit Euler method to magnetoquasistatic field problems discretized in space is given in chapter 5.

4 Solving Large Systems of Algebraic Equations

Two algorithms for solving linear systems are used in this thesis: The preconditioned conjugate gradient (PCG) method and the (sparse) LU-decomposition. Since both methods were not in any way modified or specially adapted to the work presented within this thesis, both methods are merely introduced in general in this chapter.

4.1 The Conjugate Gradient Method

The conjugate gradient method was originally published in [105] and is also well presented in [95, 108, 86, 43]. The eventually preconditioned conjugate gradient method is a Krylov subspace method for iteratively solving large sparse systems of linear algebraic equations of the form $\mathbf{Ax} = \mathbf{b}$, where the system matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ is symmetric and positive definite. Without preconditioning, the conjugate gradient method converges after n iterations at maximum. Using a suitable preconditioner improves the system matrix's condition and yields a faster convergence in less than n iterations. Furthermore, the conjugate gradient method's complexity depends on the preconditioner used.

In the following, the algorithm for the conjugate gradient method is presented first. Afterwards, the changes in the algorithm, when using a preconditioner, are explained.

The main idea underlying the conjugate gradient method is to minimize the equivalent quadratic form

$$f(\mathbf{x}) = \frac{1}{2} \mathbf{x}^\top \mathbf{Ax} - \mathbf{x}^\top \mathbf{b}, \quad \mathbf{x} \in \mathbb{R}^n, \quad (4.1)$$

instead of directly solving the system of algebraic equations $\mathbf{Ax} = \mathbf{b}$. The gradient

$$\nabla f(\mathbf{x})|_{\mathbf{x}_k} = -\mathbf{r}_k = \mathbf{Ax}_k - \mathbf{b}, \quad (4.2)$$

where \mathbf{r}_k is k -th residual, is now minimized in direction \mathbf{d}_k in a subspace V_k .

At this, the directions \mathbf{d}_k are conjugate with respect to \mathbf{A} , which means

$$\mathbf{d}_i^\top \mathbf{A} \mathbf{d}_j = 0, \quad \forall i \neq j. \quad (4.3)$$

The subspace V_k is spanned by the vectors \mathbf{d}_k as follows:

$$V_k := \mathbf{x}_0 + \text{span}\{\mathbf{d}_0, \mathbf{d}_1, \dots, \mathbf{d}_{k-1}\}. \quad (4.4)$$

At this, \mathbf{x}_0 is the initial solution, which is often chosen as $\mathbf{x}_0 = \mathbf{0}$.

The resulting algorithm comprises the steps listed in Algorithm 1.

Algorithm 1 Solve $\mathbf{Ax} = \mathbf{b}$ using the Conjugate Gradient Method

Require: $\mathbf{x}_0 \in \mathbb{R}^n$, set tolerance tol

$\mathbf{r}_0 \leftarrow \mathbf{b} - \mathbf{Ax}_0$

$\mathbf{d}_0 \leftarrow \mathbf{r}_0$

$k \leftarrow 0$

while $\|\mathbf{r}_k\| > tol$ **do**

$\mathbf{z}_k \leftarrow \mathbf{A} \mathbf{d}_k$

$\alpha_k \leftarrow (\mathbf{r}_k^\top \mathbf{r}_k) / (\mathbf{d}_k^\top \mathbf{z}_k)$

$\mathbf{x}_{k+1} \leftarrow \mathbf{x}_k + \alpha_k \mathbf{d}_k$

$\mathbf{r}_{k+1} \leftarrow \mathbf{r}_k - \alpha_k \mathbf{z}_k$

$\beta_k \leftarrow (\mathbf{r}_{k+1}^\top \mathbf{r}_{k+1}) / (\mathbf{r}_k^\top \mathbf{r}_k)$

$\mathbf{d}_{k+1} \leftarrow \mathbf{r}_{k+1} + \beta_k \mathbf{d}_k$

$k \leftarrow k + 1$

end while

For reducing the number of iterations until convergence, a preconditioner can be used. Preconditioners modify the system matrix \mathbf{A} and improve its condition number. Various preconditioners are known, such as e.g. the SSOR-preconditioner or the Jacobi preconditioner. Using the incomplete Cholesky factorization as preconditioner yields a method known as ICCG, which is very popular for application with edge element

based FEM, see e.g. [37]. Within the work of this thesis, algebraic multigrid (AMG) preconditioners are used as described e.g. in [53, 106, 9, 109]. AMG is here used as inner (preconditioner) method and requires one call per CG iteration.

The application of preconditioners in the CG method are e.g. explained in [108, 86, 22].

4.2 The Sparse LU-Decomposition

Whilst the PCG method is well suited for solving large sparse systems of linear algebraic equations, this does not hold for the sparse LU-decomposition with $\mathcal{O}(n^2)$ complexity, where n is the number of unknowns [89]. Efficient techniques for parallelizing the LU-decomposition, e.g. by using special pivoting techniques minimizing interprocessor data communication, are a still ongoing research topic, see e.g. [38, 84, 107, 44, 102, 21].

The LU-decomposition, also known as Gauss elimination, converts an equation system $\mathbf{Ax} = \mathbf{b}$ to be solved into an equivalent triangular system by linearly combining the equations. The system matrix $\mathbf{A} = \mathbf{LU}$ is decomposed into a unit lower triangular matrix \mathbf{L} and an upper triangular matrix \mathbf{U} . Thus, the system $\mathbf{Ax} = \mathbf{b}$ can be restated as

$$\mathbf{Ax} = \mathbf{b} \Leftrightarrow \mathbf{LUx} = \mathbf{Ly} = \mathbf{b}, \quad (4.5)$$

where $\mathbf{Ux} = \mathbf{y}$. Solving the equation system (4.5) comprises two steps:

1. Solve $\mathbf{Ly} = \mathbf{b}$ for \mathbf{y} ,
2. Solve $\mathbf{Ux} = \mathbf{y}$ for \mathbf{x} .

Thus, the LU-decomposition is a method suitable for multiple right-hand side (MRHS) problems, as the matrices \mathbf{L} and \mathbf{U} only depend on the system matrix \mathbf{A} and can thus be reused independently from the right-hand side.

5 (Semi-) Explicit Time Integration of the Magnetoquasistatic Field Problem

In this chapter, various numerical methods for accelerating the (semi-) explicit time integration of magnetoquasistatic field problems are presented. At this, the main idea is to use the approach originally presented in [14, 92], and to omit regularizing the singular curl-curl operator in nonconducting media by applying a Moore-Penrose pseudo-inverse. For ease of readability, it is often written "the" pseudo-inverse in the following. However, it should always be kept in mind, that more than one pseudo-inverse exists.

5.1 Application of the Moore-Penrose Pseudo-Inverse

The nonlinear magnetoquasistatic field problem (2.11) described using the magnetic vector potential \vec{A} is spatially discretized either with FIT [16] or with FEM (see equation (3.13)) and this yields the discrete formulation

$$\mathbf{M} \frac{d}{dt} \mathbf{a} + \mathbf{K}(\mathbf{a}) \mathbf{a} = \mathbf{j}_s. \quad (5.1)$$

Here, \mathbf{a} is a vector of degrees of freedom representing the time-dependent magnetic vector potential, \mathbf{M} is the mass-matrix of conductivities and \mathbf{K} is the stiffness-matrix of reluctivities. Thus, \mathbf{K} is the discretized (singular) curl-curl operator. The transient source current density is denoted by \mathbf{j}_s . At this, $\mathbf{j}_s = \mathbf{x}_s i_s(t)$ holds, where the time-dependent source current $i_s(t)$ is distributed in the spatial domain of the exciting coil

by \mathbf{x}_s [93]. For solving (5.1), initial values and boundary conditions must be imposed. Since these are subject to the specific problem considered, they are not specified here. Furthermore, the explicit reference to the time dependence of the vectors $\mathbf{a}(t)$ and $\mathbf{j}_s(t)$ is omitted for ease of readability.

Equation (5.1) represents a system of differential algebraic equations of index 1 (DAE1), since the mass-matrix of conductivities \mathbf{M} is singular [72]. The (differential) index of differential algebraic equations represents the number of differentiations with respect to time required to transform a differential algebraic equation into an ordinary differential equation [45]. Solving differential algebraic equations is usually more difficult than solving ordinary differential equations, if differentiation indices higher than 1 are considered.

As stated in the previous chapter, the common approach for integration of (5.1) is to use an implicit time integration scheme, directly applicable to systems of differential algebraic equations of index 1.

Reformulating equation (5.1) by using a Schur complement was originally proposed in [14] and reused in [92], allowing the application of a semi-explicit time integration scheme.

To highlight this approach, a generalized and simplified sketch of the computational domain Ω of magnetoquasistatic field problems is shown in Figure 5.1. Here, the computational domain Ω consists of a subdomain Ω_c of conducting material, i.e., the region where eddy currents can be excited, and of a subdomain Ω_n of nonconducting material, e.g. air (or vacuum), and of a subdomain Ω_{sc} . The subdomain Ω_{sc} comprises excitation sources, e.g. source coils. Since it is often assumed that no eddy currents are excited within Ω_{sc} , Ω_{sc} is also considered to belong to the region of nonconducting material in the following.

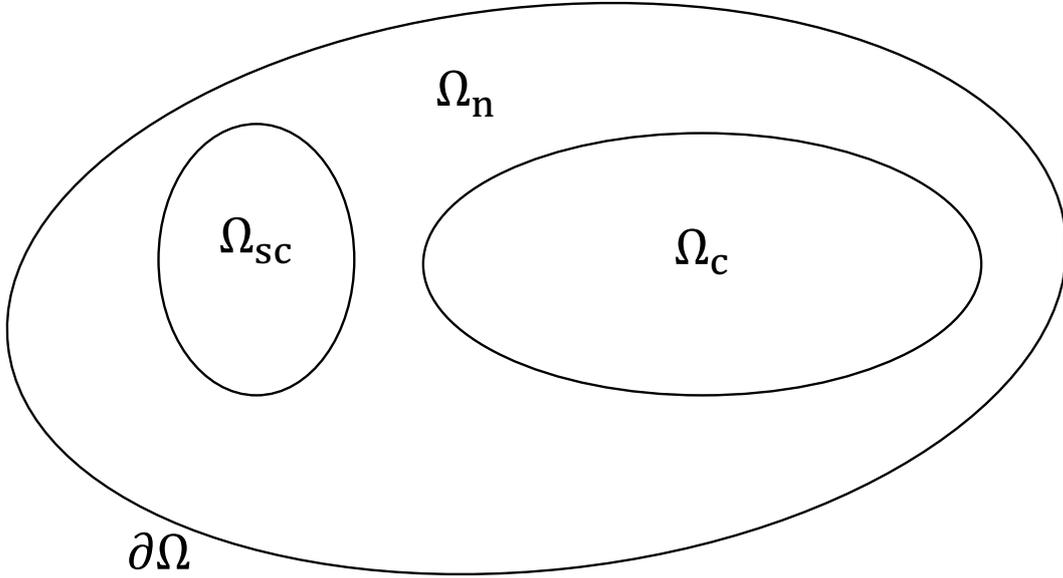


Figure 5.1: The computational domain Ω and its boundary $\partial\Omega$ comprise three subdomains: The nonconducting region (Ω_n), which has a constant permeability, the conducting region (Ω_c), where eddy currents can be excited and the permeability might be nonlinear and a nonconducting region (Ω_{sc}), where no eddy currents are excited and the excitation is placed.

By sorting and reallocating the degrees of freedom in $\mathbf{a}(t)$ according to whether they belong either to elements in Ω_c or to those in Ω_n and Ω_{sc} , equation (5.1) can also be written as

$$\begin{bmatrix} \mathbf{M}_{cc} & 0 \\ 0 & 0 \end{bmatrix} \frac{d}{dt} \begin{bmatrix} \mathbf{a}_c \\ \mathbf{a}_n \end{bmatrix} + \begin{bmatrix} \mathbf{K}_{cc}(\mathbf{a}_c) & \mathbf{K}_{cn} \\ \mathbf{K}_{cn}^\top & \mathbf{K}_{nn} \end{bmatrix} \begin{bmatrix} \mathbf{a}_c \\ \mathbf{a}_n \end{bmatrix} = \begin{bmatrix} 0 \\ \mathbf{j}_{s,n} \end{bmatrix}, \quad (5.2)$$

whereupon homogeneous Dirichlet boundary conditions are assumed. Here, \mathbf{a}_c contains the degrees of freedom belonging to the conductive subdomain Ω_c , in which eddy currents are excited and \mathbf{a}_n contains the degrees of freedom allocated in the nonconductive

subdomains Ω_n and Ω_{sc} , and $\mathbf{j}_{s,n}$ is the source current density in nonconductive subdomains. The mass matrix \mathbf{M} and the stiffness matrix \mathbf{K} are decomposed accordingly into block-matrices. The matrix \mathbf{M} is the generally singular, positive semi-definite conductivity matrix. This singularity is caused by the zero conductivity in nonconducting regions. By assigning a negligibly small artificial conductivity to the nonconducting regions, the matrix \mathbf{M} can be regularized. If a conventional Galerkin scheme employs edge elements as test and ansatz functions, the subsystem matrix \mathbf{M}_{cc} is positive definite. This is always the case within the work of this thesis.

The matrix $\mathbf{K}_{cc}(\mathbf{a}_c)$ is the curl-curl matrix's part in the conductive subdomain and \mathbf{K}_{nn} is the singular part of the curl-curl matrix in the nonconductive subdomain. The matrix \mathbf{K}_{cn} and its transpose \mathbf{K}_{cn}^\top are coupling matrices between the nonconductive and conductive subdomains.

The second block row of equation (5.2) represents the algebraic constraint of the differential algebraic equation. Without imposing a gauging condition, there is no unique solution for \mathbf{a}_n .

In [14, 92], the singular matrix \mathbf{K}_{nn} is regularized using a grad-div-regularization, which transforms the singular matrix \mathbf{K}_{nn} into a discrete vector Laplacian operator in free space. Alternatively, a tree/cotree-gauging approach [66] could be used.

Within the work of this thesis, a pseudo-inverse of the matrix \mathbf{K}_{nn} is used, as is proposed in [25]. For this, the weak gauging property of the conjugate gradient method is used [16]. If either the matrix \mathbf{K}_{nn} is regularized and inverted or if a pseudo-inverse is applied, the second block row in equation (5.2) can be solved for

$$\mathbf{a}_n = \mathbf{K}_{nn}^+ [\mathbf{j}_{s,n} - \mathbf{K}_{cn}^\top \mathbf{a}_c], \quad (5.3)$$

where a pseudo-inverse of \mathbf{K}_{nn} is represented by \mathbf{K}_{nn}^+ .

Equation (5.3) is inserted into the first block row of equation (5.2), which yields

$$\mathbf{M}_{cc} \frac{d}{dt} \mathbf{a}_c + \mathbf{K}_{cc}(\mathbf{a}_c) \mathbf{a}_c + \mathbf{K}_{cn} [\mathbf{K}_{nn}^+ \mathbf{j}_{s,n} - \mathbf{K}_{nn}^+ \mathbf{K}_{cn}^\top \mathbf{a}_c] = 0. \quad (5.4)$$

Equation (5.4) can be reformulated as

$$\mathbf{M}_{cc} \frac{d}{dt} \mathbf{a}_c + [\mathbf{K}_{cc}(\mathbf{a}_c) - \mathbf{K}_{cn} \mathbf{K}_{nn}^+ \mathbf{K}_{cn}^\top] \mathbf{a}_c = -\mathbf{K}_{cn} \mathbf{K}_{nn}^+ \mathbf{j}_{s,n}, \quad (5.5)$$

where the matrix product $\mathbf{K}_{\text{cn}}\mathbf{K}_{\text{nn}}^+\mathbf{K}_{\text{cn}}^\top$ is the generalized Schur complement

$$\mathbf{K}_S = \mathbf{K}_{\text{cn}}\mathbf{K}_{\text{nn}}^+\mathbf{K}_{\text{cn}}^\top. \quad (5.6)$$

Due to the elimination of the algebraic part in (5.2) related to the dofs in the non-conducting domain, equation (5.5) is a system of ordinary differential equations. The presented approach thus transforms the system of differential algebraic equations described by (5.1), respectively by (5.2), of infinite stiffness, into a system of ordinary differential equations of finite stiffness. Thus, explicit time integration of (5.5) is feasible.

As stated in [45]: "While the intuitive meaning of stiff is clear to all specialists, much controversy is going on about its correct mathematical definition (see e.g. p.360-363 of Aiken (1985)). The most pragmatical opinion is also historically the first one (Curtiss & Hirschfelder 1952): stiff equations are equations where certain implicit methods, in particular BDF perform better, usually tremendously better, than explicit ones" ([45], p. 1, ll. 9-14.).

By applying the explicit Euler method for the time integration of (5.5) with a time step size Δt , the new time step solution vector

$$\mathbf{a}_c^{m+1} := \mathbf{a}_c^m + \Delta t \mathbf{M}_{\text{cc}}^{-1} \left[\mathbf{K}_{\text{cn}}\mathbf{K}_{\text{nn}}^+\mathbf{j}_{\text{s,n}}^{m+1} - \left(\mathbf{K}_{\text{cc}}(\mathbf{a}_c^m) - \mathbf{K}_{\text{cn}}\mathbf{K}_{\text{nn}}^+\mathbf{K}_{\text{cn}}^\top \right) \mathbf{a}_c^m \right], \quad (5.7)$$

is computed in the $(m+1)$ -th time step. Evaluating (5.7) updates only the solution for the degrees of freedom allocated in conducting regions \mathbf{a}_c . The solution for the degrees of freedom allocated in nonconducting regions \mathbf{a}_n can then easily be updated for the $(m+1)$ -th time step by evaluating (5.3) with

$$\mathbf{a}_n^{m+1} := \mathbf{K}_{\text{nn}}^+\mathbf{j}_{\text{s,n}}^{m+1} - \mathbf{K}_{\text{nn}}^+\mathbf{K}_{\text{cn}}^\top \mathbf{a}_c^{m+1}. \quad (5.8)$$

The computationally expensive regularization of the singular matrix \mathbf{K}_{nn} proposed in [14], can be avoided by evaluating a pseudo-inverse using a preconditioned conjugate gradient method. For evaluating a pseudo-inverse in (5.7) and (5.8), systems of the kind of

$$\mathbf{K}_{\text{nn}}\mathbf{a}_p = \mathbf{r}_p \Leftrightarrow \mathbf{a}_p = \mathbf{K}_{\text{nn}}^+\mathbf{r}_p \quad (5.9)$$

need to be solved. The explicit computation of the matrix \mathbf{K}_{nn}^+ representing a pseudo-inverse of \mathbf{K}_{nn} is not required. Instead, the matrix vector multiplication $\mathbf{K}_{\text{nn}}^+\mathbf{r}_p$ is

replaced with a vector \mathbf{a}_p , which is computed according to (5.9) for each right-hand side vector \mathbf{r}_p using the PCG method. Here, \mathbf{r}_p is a general representation for the vectors with which \mathbf{K}_{mn}^+ is multiplied with in equations (5.7) and (5.8). Thus, when solving equation (5.9) for \mathbf{a}_p , \mathbf{r}_p is either replaced by $\mathbf{K}_{cn}^\top \mathbf{a}_c^m$ or by $\mathbf{K}_{cn}^\top \mathbf{a}_c^{m+1}$. The evaluation of a pseudo inverse for $\mathbf{K}_{mn} \mathbf{a}_p = \mathbf{j}_{s,n}^{m+1}$ in equations (5.7) and (5.8), is only required once, since the vector of the current density $\mathbf{j}_{s,n}^{m+1}$ is updated in each time step by multiplication with a scalar value. This is explained in more detail in subchapter 5.2.

The maximum stable time step size of explicit time integration methods is estimated by the Courant-Friedrich-Levy (CFL) criterion [19, 45]. Here,

$$\Delta t \leq \frac{2}{\lambda_{\max}(\mathbf{M}_c^{-1}(\mathbf{K}_{cc}(\mathbf{a}_c^m) - \mathbf{K}_S))} \quad (5.10)$$

can be used to estimate the maximum stable time step size. At this, λ_{\max} denotes the maximum eigenvalue and can be evaluated using the power method [92, 43].

5.1.1 Numerical Validation: Application of a Pseudo-Inverse

The results in this section are originally presented in [25].

All methods for the (semi-) explicit time integration of the magnetoquasistatic field problem described in this thesis are coded using the code package "MEQSICO" as platform. "MEQSICO" is a C++ code for the numerical simulation of three-dimensional electroquasistatic and magnetoquasistatic field problems [15, 104]. The implementation of the finite element method in MEQSICO is based on the Femster library [11, 12]. First order edge-elements are used for spatial discretization [72].

For preconditioning the conjugate gradient method, algebraic multigrid preconditioners from the Trilinos ML package of the Trilinos Project are used [42, 51, 63, 106]. All simulations are run on server with an Intel Xeon E5-2660 processor and an Nvidia K80 GPU.

In the following, two benchmark problems from the "TEAM Benchmark Suite" of the Compumag Society and their numerical solution using the MEQSICO code and the application of a pseudo-inverse for semi-explicit time integration are presented.

5.1.1.1 Results for the TEAM 7 benchmark problem

The linear "TEAM 7" problem is a three-dimensional model problem consisting of a current source coil above an aluminum plate with a rectangular hole in it, as shown in Fig. 5.2. The problem description and reference solutions are published in [36]. A side view of the problem's geometry is shown on the left side of Figure 5.2. The aluminum plate is colored in blue and the exciting source coil is colored in grey. The aluminum plate corresponds to the conducting subdomain Ω_c . The source coil and the air gap, as well as the air surrounding the plate and the coil, belong to the nonconducting subdomain Ω_n .

For simulating the problem, plate and coil are surrounded by an air box. Homogeneous Neumann boundary conditions and homogeneous Dirichlet boundary conditions for the magnetic vector potential \vec{A} are defined on the surfaces of this air box.

The magnetic flux density is evaluated along the line P drawn in orange. A top view of the model geometry is shown on the right side of Figure 5.2. The excitation current in the source coil is described by a ramped sinus function with a frequency of 50 Hz. The simulations are executed for a time interval ranging from 0 s to 60 ms, thus the first three periods are simulated. The source current is shown in Figure 5.3.

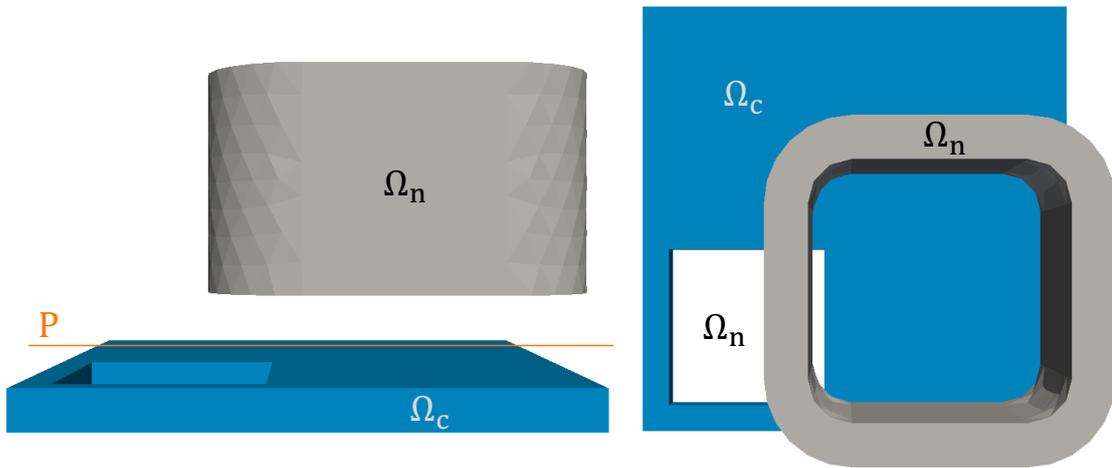


Figure 5.2: On the left side: Model geometry of the TEAM 7 benchmark problem side view. The magnetic flux density is evaluated along the orange line P.

On the right side: Top view of the model geometry of the TEAM 7 benchmark problem.

The model problem is meshed with two different tetrahedral meshes, as shown in Figure 5.4. The coarse mesh results in 46,218 degrees of freedom and is shown on the left side of Figure 5.4. The fine mesh results in 247,276 degrees of freedom and is shown on the right side of Figure 5.4. For each mesh, two simulations are executed: A first simulation using the implicit Euler method for time integration and a second simulation employing the explicit Euler scheme and the application of a pseudo-inverse

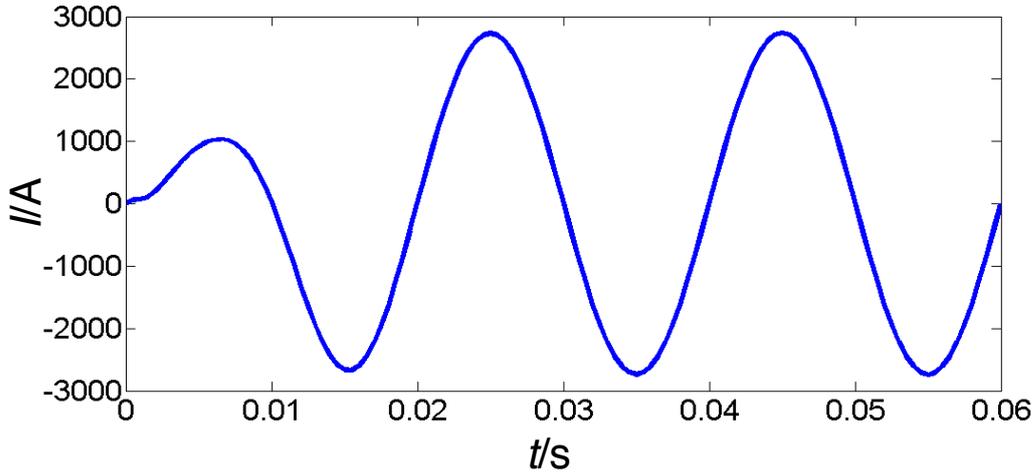


Figure 5.3: Excitation current flowing in the source coil of the TEAM 7 benchmark problem

of the matrix \mathbf{K}_{nn} .

According to the Courant-Friedrichs-Lewy criterion, the maximum stable time step size for the coarsely meshed problem is $\Delta t_{\text{CFL}} = 2.95 \cdot 10^{-4}$ s and for the problem using a fine mesh $\Delta t_{\text{CFL}} = 8.02 \cdot 10^{-5}$ s.

The numerical simulations of the linear TEAM 7 problem were the first simulations using the explicit Euler method and the application of a pseudo-inverse carried out within the work for this thesis. Therefore, the smaller time step sizes than the maximum stable time step sizes were used to avoid any stability issues. Thus, the coarsely meshed problem is simulated using an explicit time step size $\Delta t_{\text{exp}} = 1.0 \cdot 10^{-4}$ s and the fine meshed problem is simulated using an explicit time step size $\Delta t_{\text{exp}} = 5.0 \cdot 10^{-5}$ s. The time step size for the reference simulations using the implicit Euler method is $\Delta t_{\text{imp}} = 1.0 \cdot 10^{-4}$ s for both meshes. The accuracy of the PCG solver for evaluating a pseudo-inverse of

\mathbf{K}_{nn} is set to $\text{tol} = 10^{-16}$.

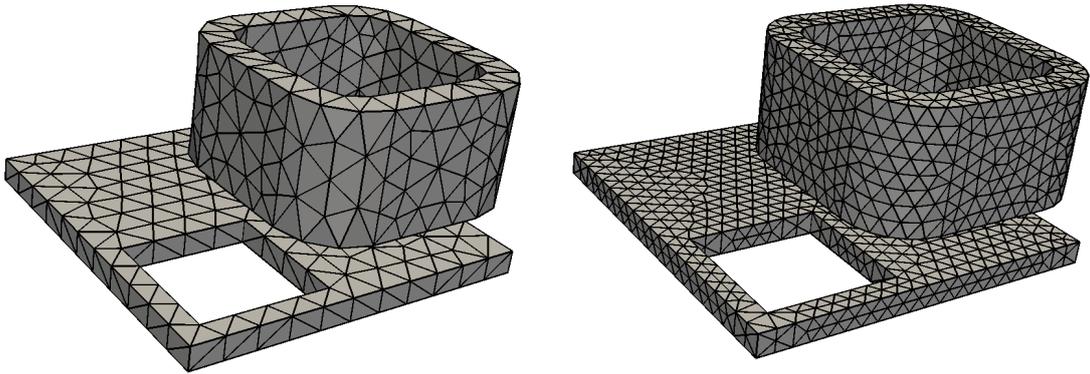


Figure 5.4: On the left side: Mesh resulting in 46,218 degrees of freedom.

On the right side: Mesh resulting in 247,276 degrees of freedom.

The results for the magnitude of the B_z -component of the magnetic flux density evaluated along the line P are shown in Figure 5.5. Here, the results for the fine mesh obtained using the implicit Euler method or the explicit Euler method are compared with the measurement results published in [36].

As shown in Figure 5.5, the simulation results obtained from using the explicit and the implicit Euler scheme are in good agreement. All results depicted are based on the fine mesh. This can be considered as proof of concept for the proposed application of a pseudo-inverse. The simulation results slightly deviate from the measurement results. This can be due to the fine mesh still not being sufficiently fine in the gap between plate and coil.

Field plots of the magnetic flux density and the excited eddy current density are shown

in Figure 5.6. Here, the fields shown on the left side are computed using the implicit Euler method and the fields on the right side are evaluated using the explicit Euler method.

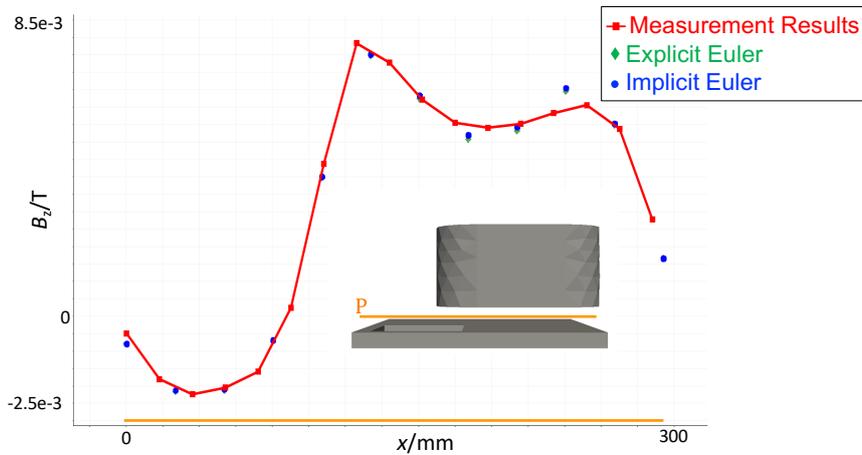


Figure 5.5: Results for the magnitude of the B_z -component of the magnetic flux density along the evaluation line P computed using the explicit or the implicit Euler method, in comparison with the measurement results published in [36]. The results are obtained using the fine mesh resulting in 247,276 degrees of freedom.

The resulting simulation times are shown in Table 5.1. It should be noted, that both explicit and implicit simulation algorithms are not in any way optimized in these simulations. The simulations based on the coarse mesh use the same time step size for the explicit and the implicit time integration and both yield the same simulation time of 12 min. This shows that the computational effort for an explicit and an implicit time step are the same for this code version. This is no surprise, as a linear material is used in the simulations, thus no linearization is required within an im-

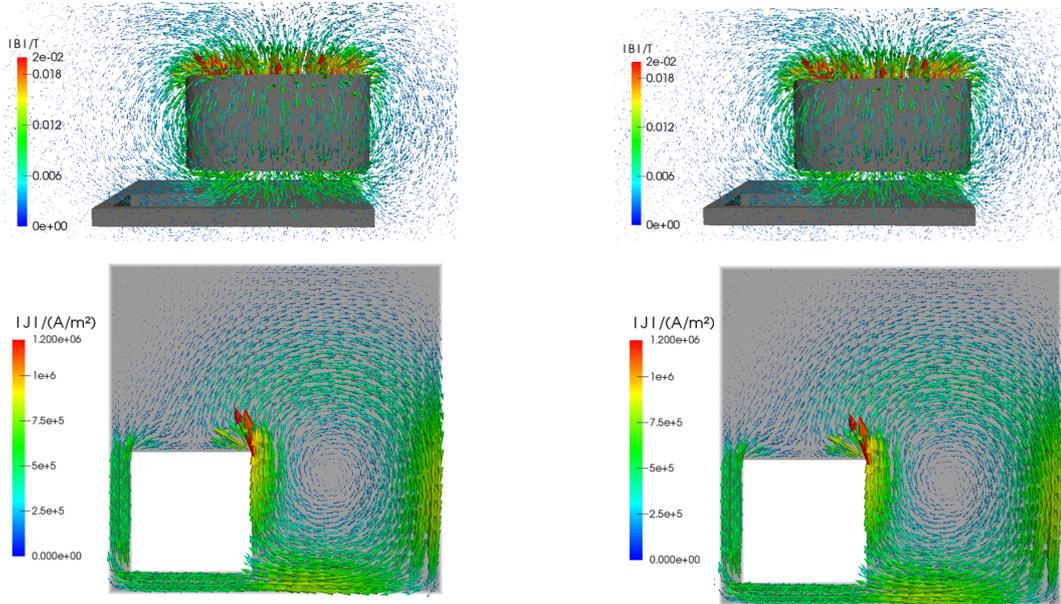


Figure 5.6: On the top left side: Magnetic flux density computed using the implicit Euler method.

On the bottom left side: Eddy currents computed using the implicit Euler method.

On the top right side: Magnetic flux density computed using the explicit Euler method.

On the bottom right side: Eddy currents computed using the explicit Euler method.

All results are based on the fine mesh resulting in 247,276 degrees of freedom.

implicit time step. The simulations on the fine mesh use half the time step size, i.e., $\Delta t_{\text{exp}} = 0.5 \cdot 10^{-4}$ s for the explicit time integration than for the implicit time integration with $\Delta t_{\text{imp}} = 1.0 \cdot 10^{-4}$ s. Accordingly, it takes almost twice the simulation time for the explicit time integration than for the implicit time integration.

However, these simulations were only performed to prove that the application of a

pseudo-inverse is a valid approach. It must also be noted, that the problem considered does not comprise a nonlinear ferromagnetic material, thus, no linearization is required in the implicit time integration.

It can also be seen, that the small stable time step sizes for explicit time integration schemes, related to small spatial mesh elements, are a general drawback that must be compensated for using the advantages resulting from explicit time integration schemes. This will be shown in more detail in later chapters of this thesis.

Time Integration Method	# dofs	Time Step Size	Simulation Time
Implicit Euler	46,218	$1.0 \cdot 10^{-4} s$	12 min
Explicit Euler	46,218	$1.0 \cdot 10^{-4} s$	12 min
Implicit Euler	247,276	$1.0 \cdot 10^{-4} s$	2.5 h
Explicit Euler	247,276	$0.5 \cdot 10^{-4} s$	4.75 h

Table 5.1: Time step sizes and resulting simulation times for all simulations of the TEAM 7 benchmark problem carried out. All simulations are run on a server with an Intel Xeon E5-2660 processor.

5.1.1.2 Results for the TEAM 10 benchmark problem

Another problem evaluated is the TEAM 10 benchmark problem. The TEAM 10 problem is a three-dimensional model problem comprising nonlinear ferromagnetic steel plates and narrow air gaps. The problem description and reference solutions are presented in [70]. Its geometry is shown in Figure 5.7. Here, the steel plates are colored in blue and red. The exciting source coil is colored in green. Both air gaps between the blue and red steel plates are of the same width $d_g = 0.5$ mm. The steel plates belong to the conducting subdomain Ω_c . The exciting source coil, the air gap, and the air surrounding the shown geometry belong to the nonconducting subdomain Ω_n .

The depicted geometry lies within a box shaped computational domain on which's surfaces homogeneous Neumann boundary conditions and homogeneous Dirichlet boundary conditions for the magnetic vector potential \vec{A} are defined in the simulation. This box is not shown in Figure 5.7. The lines S1, S2, and S3 in Figure 5.7 mark the positions where the average magnetic flux density is measured, as described in [70].

The transient excitation current in the source coil is described by a function $i_s = (1 - \exp(-t/\tau))$. Here, $\tau = 0.05$ s is a time constant. The current starts to flow at time $t = 0$ s and the resulting magnetic flux density is simulated for the following 150 ms. The excitation current is shown in Figure 5.8 and the calculated magnetization curve of the steel plates is shown in Figure 5.9.

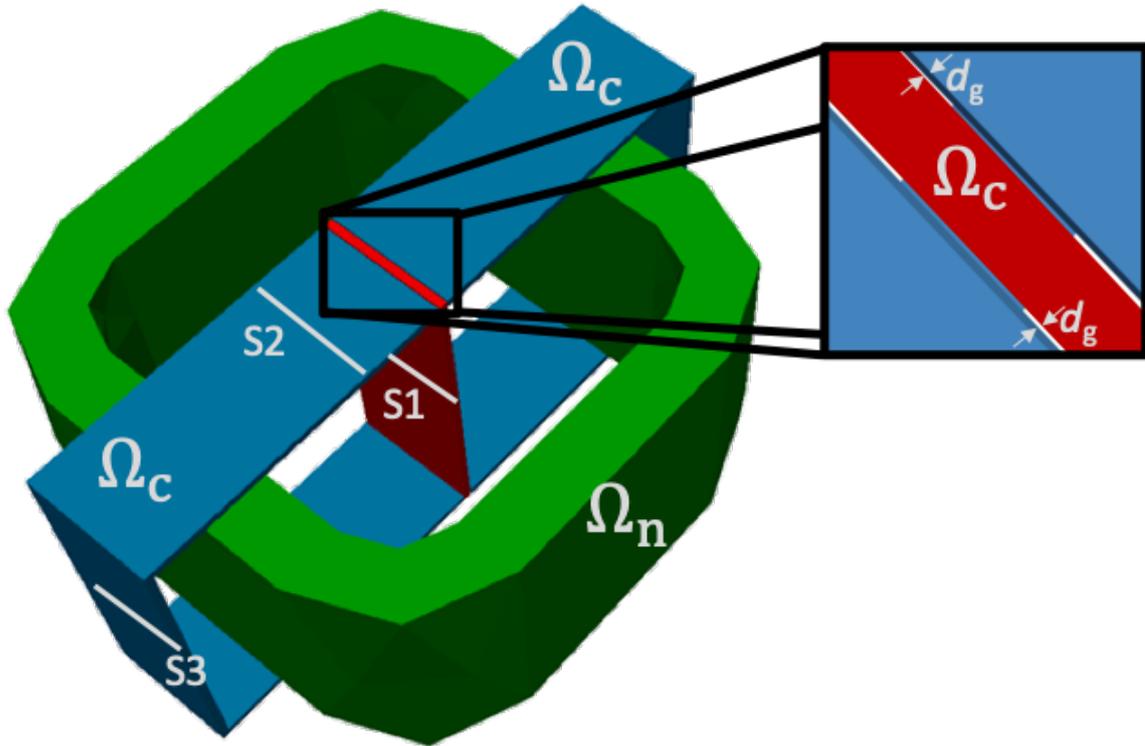


Figure 5.7: Model geometry of the TEAM 10 benchmark problem.

Two different tetrahedral element meshes of different resolution are applied to the TEAM 10 problem. The coarse mesh shown on the left side of Figure 5.10 yields 29,532 degrees of freedom. The fine mesh depicted on the right side of Figure 5.10 results in 130,098 degrees of freedom.

As for the TEAM 7 problem, both a simulation using the implicit Euler method and another simulation using the explicit Euler method for time integration are carried out for each mesh respectively. For the implicit time integration scheme a time step size $\Delta t_{\text{imp}} = 2$ ms is chosen for both mesh resolutions.

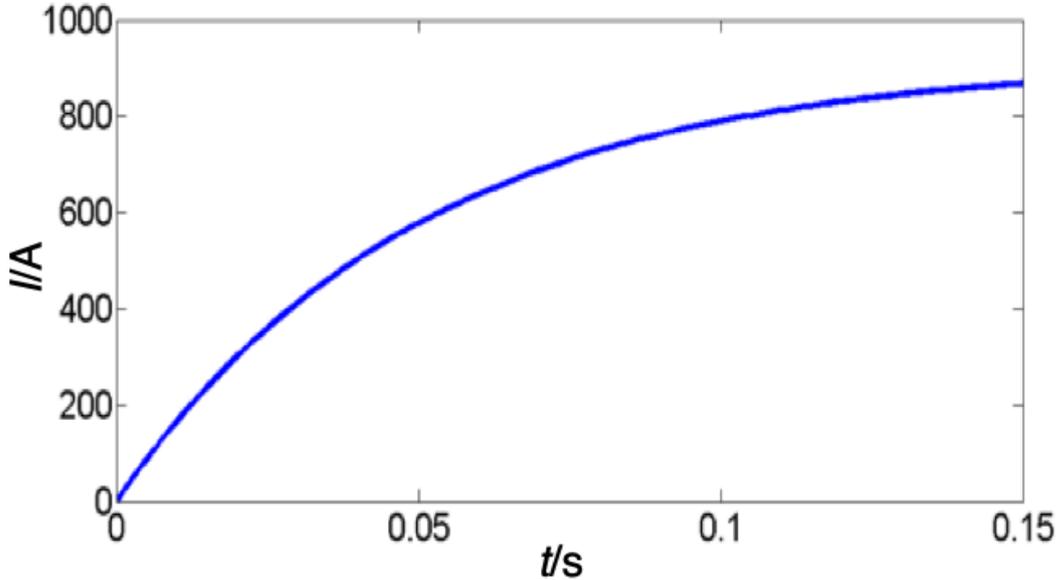


Figure 5.8: Current flowing in the exciting source coil of the TEAM 10 benchmark problem. The current is described with the function $i_s = (1 - \exp(-t/\tau))$, where $\tau = 0.05$ s.

The Courant-Friedrichs-Lewy criterion states a maximum stable time step size $\Delta t_{\text{CFL}} = 1.85 \cdot 10^{-6}$ s for the coarsely meshed problem and $\Delta t_{\text{CFL}} = 3.0 \cdot 10^{-7}$ s for the fine meshed problem. These maximum stable time step sizes are evaluated, prior to executing the simulations, using the Power Method [43] for computing the maximum eigenvalue in (5.10), where, the maximum value for the permeability μ of the magnetization curve of the TEAM 10 problem is used to obtain a "worst-case" time steps size that is stable for all values of the permeability μ .

Just as in case of the numerical simulations of the TEAM 7 problem, smaller time step sizes than the maximum stable time step sizes were used to avoid any stability issues. Thus, the coarsely meshed problem is simulated using an explicit time step size $\Delta t_{\text{exp}} = 1.0 \cdot 10^{-6}$ s and the fine meshed problem is simulated using an explicit time step size $\Delta t_{\text{exp}} = 1.0 \cdot 10^{-7}$ s.

The resulting field plots for the eddy currents evaluated on the coarse and on the fine mesh are shown in Figure 5.11.

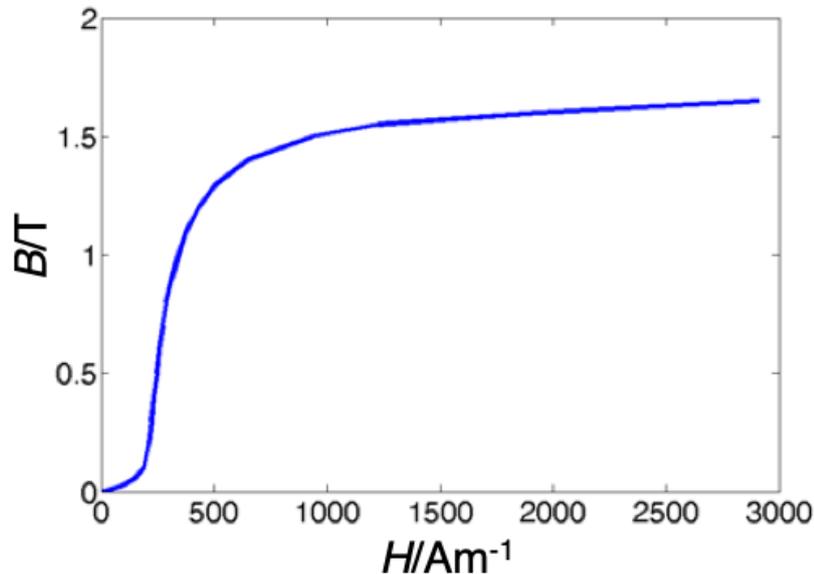


Figure 5.9: Ferromagnetic nonlinear B - H -curve of the steel plates according to the material data published in [70].

The resulting field plots for the magnetic flux density evaluated on the coarse mesh are shown in Figure 5.12. On the left side of Figure 5.12, the resulting field plot from applying the implicit Euler scheme for time integration is shown. On the right side of Figure 5.12, the resulting field plot from applying the explicit Euler scheme for time integration is depicted. Already from these field plots, a slight deviation in the solution obtained using the implicit and the explicit time integration method is apparent. The same holds for Figure 5.13 in which the magnitude of the z -component of the magnetic flux density evaluated along line S1 is shown. Here, the implicit and the explicit solution also slightly deviate from each other.

This phenomenon is further investigated and explained in chapter 5.3. Here, it is only briefly mentioned, that in contrast to the TEAM 7 problem with a constant magnetic permeability, a strongly nonlinear permeability is considered here. This means, in the implicit time integration algorithm linearizations are performed. Each linearization is stopped, if a certain tolerance is reached. The explicit time integration scheme, however, does not require such a linearization. Thus, as will be focused on in chapter 5.3, the simulation using an explicit time integration method yields more accurate

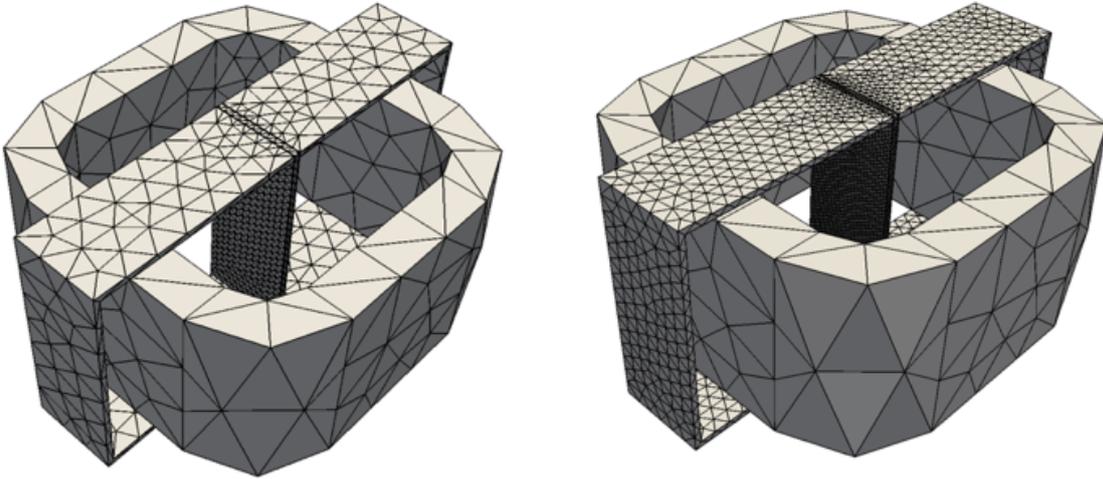


Figure 5.10: On the left side: Coarsely meshed TEAM 10 problem, yielding 29,532 degrees of freedom.

On the right side: Fine meshed TEAM 10 problem, yielding 130,098 degrees of freedom.

results. This causes the deviation to the implicit solution.

Both, the implicit and the explicit solution, deviate from the measurement results published in [70]. This is due to the insufficient resolution of the problem geometry by a coarse mesh. The simulation times required for simulating the fine mesh are extremely long, as can be seen in Table 5.1. Therefore, only a time interval ranging from 0 s to 10 ms is simulated for the fine mesh. The time step size for the implicit Euler method on the fine mesh is reduced to $\Delta t_{\text{imp}} = 1.0 \cdot 10^{-4}$ s, as a time step size of $\Delta t_{\text{imp}} = 2.0 \cdot 10^{-3}$ s with the employed code, results in more than thirty Newton-Raphson iterations per implicit time step. In order to reduce the number of Newton-Raphson iterations, the time step size is reduced. However, it should be noted, that the coding of the Newton-Raphson method does not involve a relaxation with an optimized relaxation value. This will certainly reduce the number of required Newton-Raphson iterations, if a larger implicit time step size is used.

The resulting simulation times are shown in Table 5.2. It should again be noted, that

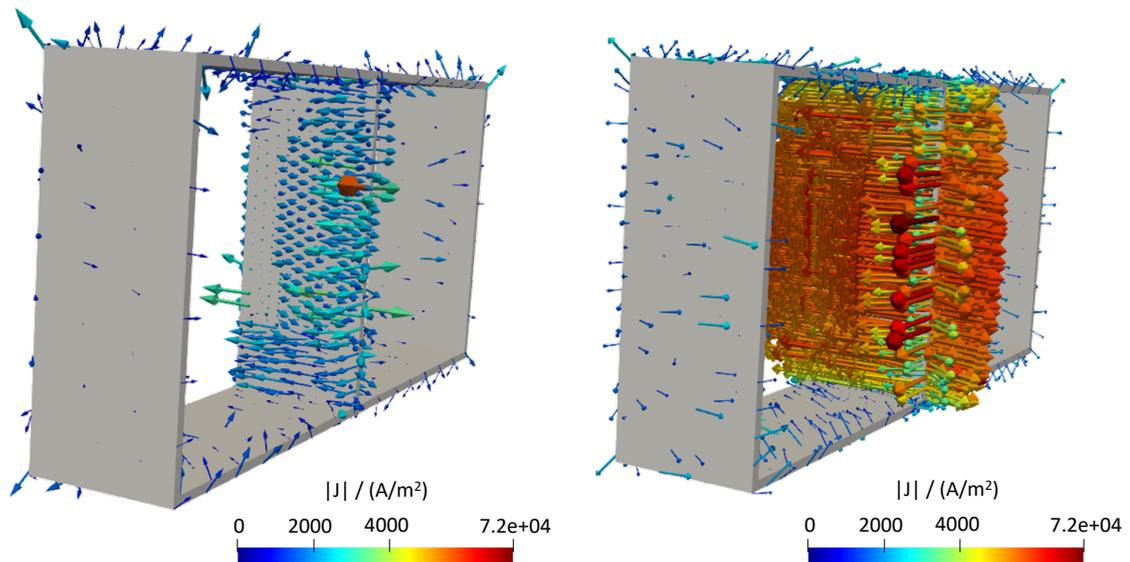


Figure 5.11: On the left side: Field plot of the eddy currents evaluated on the coarse mesh, using the explicit Euler method for time integration.
On the right side: Field plot of the eddy currents evaluated on the fine mesh, using the explicit Euler method for time integration.

both explicit and implicit simulation algorithms are not in any way optimized with respect to computational efficiency and speed in these simulations.

The resulting simulation times clearly demonstrate that the application of an explicit time integration method does not automatically yield a speed-up and that further improvements are required.

Yet, in case of the coarse mesh, the explicit time step size is 2000 times smaller than the implicit time step size for the given mesh, but the explicit simulation time is only 54 times longer than the implicit simulation time. This shows, that the computational effort for an explicit time step is much smaller. This also holds for the simulations on the fine mesh. Here, the explicit time step is 1000 times smaller than the implicit time step, but the explicit simulation time is only 180 times longer.

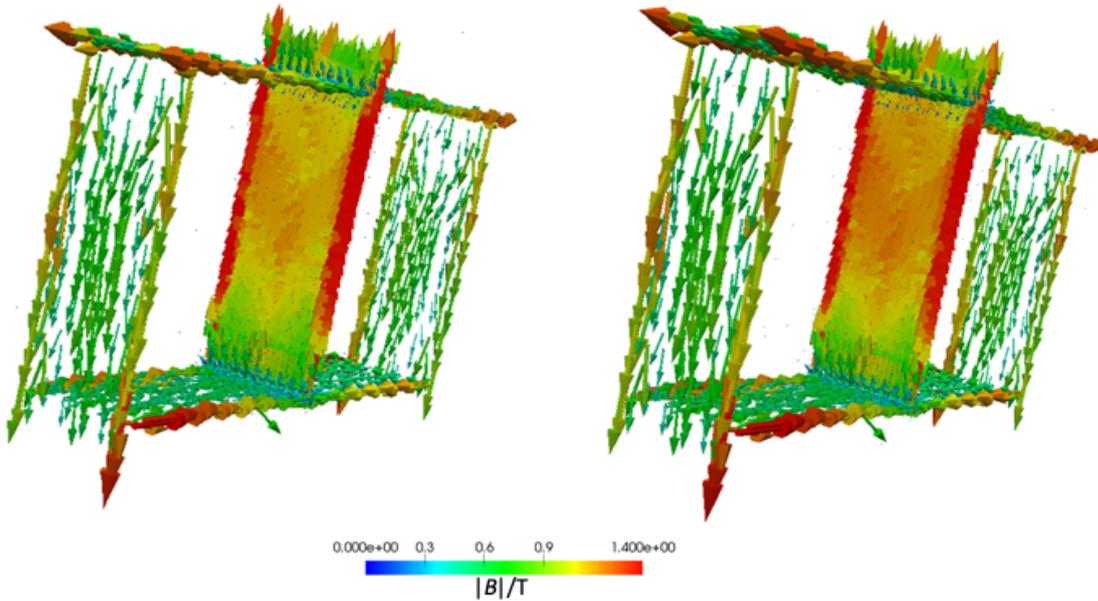


Figure 5.12: On the left side: Field plot of the magnetic flux density evaluated on the coarse mesh, using the implicit Euler method for time integration.
On the right side: Field plot of the magnetic flux density evaluated on the coarse mesh, using the explicit Euler method for time integration.

Time Integration Method	# dofs	Time Step Size	Time Interval	Simulation Time
Implicit Euler	29,532	$2.0 \cdot 10^{-3} s$	150 ms	57 min
Explicit Euler	29,532	$1.0 \cdot 10^{-6} s$	150 ms	51 h
Implicit Euler	130,098	$1.0 \cdot 10^{-4} s$	10 ms	2 h 17 min
Explicit Euler	130,098	$1.0 \cdot 10^{-7} s$	10 ms	15 d

Table 5.2: Time step sizes and resulting simulation times for all simulations of the TEAM 10 benchmark problem carried out. All simulations are run on a server with an Intel Xeon E5-2660 processor.

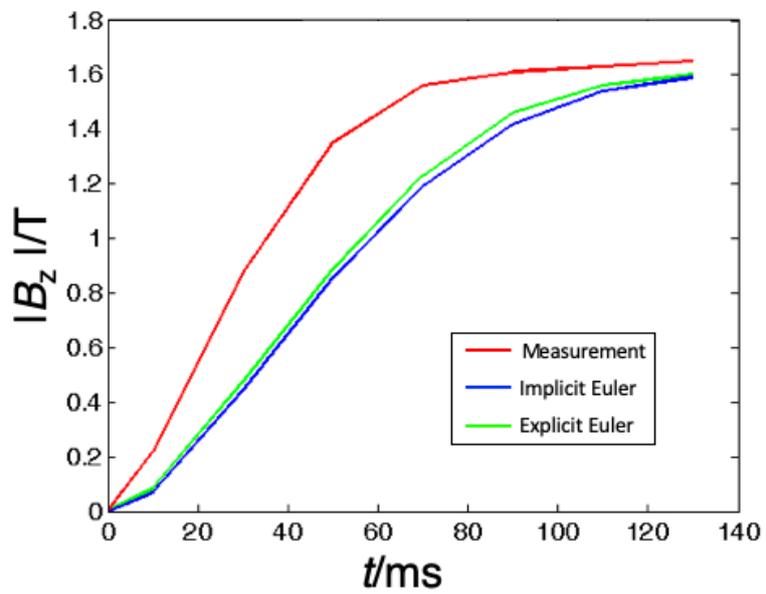


Figure 5.13: Results on the coarse grid for the magnitude of the z-component of the magnetic flux density evaluated along line S1.

5.2 Improved Start Vector Generation Within Semi-Explicit Time Stepping

Updating the vector of the degrees of freedom in conducting material in the $(m+1)$ -th time step according to equation (5.7), requires solving systems of equations. The mass matrix \mathbf{M}_{cc} is inverted by solving an equation system of the type

$$\mathbf{M}_{cc}\mathbf{x} = \mathbf{r}^{m+1} \Leftrightarrow \mathbf{x} = \mathbf{M}_{cc}^{-1}\mathbf{r}^{m+1}, \quad (5.11)$$

where $\mathbf{r}^{m+1} = \left[\mathbf{K}_{cn}\mathbf{K}_{nn}^+\mathbf{j}_{s,n}^{m+1} - \left(\mathbf{K}_{cc}(\mathbf{a}_c^m) - \mathbf{K}_{cn}\mathbf{K}_{nn}^+\mathbf{K}_{cn}^\top \right) \mathbf{a}_c^m \right]$ is newly evaluated in each explicit Euler method time step.

Here, instead of evaluating a pseudo-inverse by solving

$$\mathbf{K}_{nn}\mathbf{a}_p = \mathbf{j}_{s,n}^{m+1} \Leftrightarrow \mathbf{a}_p = \mathbf{K}_{nn}^+\mathbf{j}_{s,n}^{m+1} \quad (5.12)$$

for \mathbf{a}_p in each time step, the evaluation of a pseudo-inverse for $\mathbf{K}_{nn}\mathbf{a}_p = \mathbf{j}_{t0,1}$, is only executed once at the beginning t_0 of the time stepping. As the current density $\mathbf{j}_s(t) := f(t) \cdot \mathbf{j}_{t0,1}$, ($f: \mathbb{R} \rightarrow \mathbb{R}$) is the product of a time-dependent scalar value $f(t)$ and a constant vector $\mathbf{j}_{t0,1}$ pointing in the direction of current flow, the pseudo-inverse can be updated by multiplication with the scalar value $f(t)$ in each time step, i.e., $\mathbf{a}_p = f(t) \cdot \mathbf{K}_{nn}^+\mathbf{j}_{t0,1}$.

For computing the product $\mathbf{K}_{cn}\mathbf{K}_{nn}^+\mathbf{K}_{cn}^\top\mathbf{a}_c^m$ in equation (5.7), a pseudo-inverse is evaluated by solving a system of the kind of

$$\mathbf{K}_{nn}\mathbf{a}_p = \mathbf{K}_{cn}^\top\mathbf{a}_c^m \Leftrightarrow \mathbf{a}_p = \mathbf{K}_{nn}^+\mathbf{K}_{cn}^\top\mathbf{a}_c^m. \quad (5.13)$$

Both systems, represented by equations (5.11) and (5.13), are solved by the preconditioned conjugate gradient method, which is well suited for solving large systems of equations comprising sparse matrices.

The preconditioned conjugate gradient method is an iterative method, as outlined in chapter 4.1. By supplying it with an optimal start vector, convergence can be significantly improved resulting in fewer iterations and thus less computation time.

In the following, two methods for computing such optimal start vectors are presented. Since the entries of the matrices \mathbf{M}_{cc} and \mathbf{K}_{nn} do not change over time, the equations

(5.11) and (5.13) form multiple right-hand side (MRHS) problems in which only the right-hand side changes.

5.2.1 Subspace Projection Extrapolation and Cascaded Subspace Projection Extrapolation

A method usable for computing such optimal start vectors for the preconditioned gradient method is the subspace projection extrapolation (SPE) method proposed in [17]. The application of the SPE method to the equation systems (5.11) and (5.13) is proposed in [29, 32].

In a first step, solution vectors for \mathbf{x}^n and \mathbf{a}_p^n from n previous time steps are orthonormalized. The modified Gram-Schmidt method can be applied for the orthonormalization [43]. The n normalized, linearly independent vectors make up the column vector of an operator $\mathbf{V}_M = \{\mathbf{v}_{M1} \mid \dots \mid \mathbf{v}_{Mn}\}$ used for determining an optimal start vector for inverting the mass matrix \mathbf{M}_{cc} , respectively, an operator $\mathbf{V}_K = \{\mathbf{v}_{K1} \mid \dots \mid \mathbf{v}_{Kn}\}$ used for determining an optimal start vector for evaluating a pseudo-inverse of \mathbf{K}_{nn} .

In a second step, the projected system

$$\mathbf{V}_M^\top \mathbf{M}_{cc} \mathbf{V}_M \mathbf{z}_M = \mathbf{V}_M^\top \mathbf{r}^{m+1} \quad (5.14)$$

is solved for $\mathbf{z}_M \in \mathbb{R}^n$ and the projected system

$$\mathbf{V}_K^\top \mathbf{K}_{nn} \mathbf{V}_K \mathbf{z}_K = \mathbf{V}_K^\top \mathbf{K}_{cn}^\top \mathbf{a}_c^m \quad (5.15)$$

is solved for $\mathbf{z}_K \in \mathbb{R}^n$. Both equation systems (5.14) and (5.15) can be solved fast using a direct method, since the matrix products $\mathbf{V}_M^\top \mathbf{M}_{cc} \mathbf{V}_M$ and $\mathbf{V}_K^\top \mathbf{K}_{nn} \mathbf{V}_K$ both yield an $n \times n$ -matrix.

The entries of the vectors \mathbf{z}_M and \mathbf{z}_K are the coefficients with which respectively a new optimal start vector

$$\mathbf{v}_{M0,SPE} := \mathbf{V}_M \mathbf{z}_M \quad (5.16)$$

and

$$\mathbf{v}_{K0,SPE} := \mathbf{V}_K \mathbf{z}_K \quad (5.17)$$

can be obtained. Thus, an optimal start vector for the preconditioned conjugate gradient method results from a special linear combination of the column vectors of the operator \mathbf{V}_M , respectively \mathbf{V}_K [17].

In the work of this thesis, a modified version of the subspace projection extrapolation method, termed the Cascaded Subspace Projection Extrapolation (CSPE) method presented in [29, 32] is employed.

The solution vectors \mathbf{x}^m and \mathbf{a}_p^m from the m -th time step are orthonormalized in the $(m + 1)$ -th time step using the modified Gram-Schmidt method against the column vectors of the operator \mathbf{V}_M and the column vectors of the operator \mathbf{V}_K , respectively. The resulting vectors are referred to as \mathbf{v}_M^{m+1} and \mathbf{v}_K^{m+1} . The column vectors \mathbf{v}_M^{m+1} and \mathbf{v}_K^{m+1} could either be appended to the column vectors of \mathbf{V}_M and \mathbf{V}_K , thus increasing the information content of both operators, or could replace the last column vector of \mathbf{V}_M and \mathbf{V}_K , respectively, thus only updating the information content.

A measure for the spectral information content stored in the operators \mathbf{V}_M and \mathbf{V}_K is the number of iterations the preconditioned conjugate gradient method needs for convergence. Thus, the spectral information content stored is directly related to the optimality of the start vector computed. If the number of required conjugate gradient iterations is increasing, i.e., if the number of iterations in the m -th time step is larger than in the $(m - 1)$ -th time step, the vector \mathbf{v}_M^{m+1} , respectively the vector \mathbf{v}_K^{m+1} is appended to the corresponding operator \mathbf{V}_M , or \mathbf{V}_K , in order to increase the operator's spectral information content.

Continuously appending column vectors to either \mathbf{V}_M , or \mathbf{V}_K , would result in increasingly expensive computations in solving (5.14) and (5.15). This is prevented by defining a maximum limit N_{CG} of an acceptable number of conjugate gradient iterations. Thus, the column vector \mathbf{v}_M^{m+1} , respectively the vector \mathbf{v}_K^{m+1} is only appended to \mathbf{V}_M , or \mathbf{V}_K respectively, if both conditions are fulfilled:

1. The number of conjugate gradient iterations in the m -th time step is larger than the number of conjugate gradient iterations in the $(m - 1)$ -th time step, AND
2. The number of conjugate gradient iterations in the m -th time step is greater than N_{CG} .

In the computation of the matrix-matrix products $\mathbf{M}_{\text{cc}}\mathbf{V}_M$ and $\mathbf{K}_{\text{nn}}\mathbf{V}_K$ in equations (5.14) and (5.15), only the products of matrix-vector $\mathbf{M}_{\text{cc}}\mathbf{v}_M^{m+1}$ and $\mathbf{K}_{\text{nn}}\mathbf{v}_K^{m+1}$ change in each time step. The remaining matrix-vector products $\mathbf{M}_{\text{cc}}\mathbf{v}_M^i, i = 1 \dots m$ and $\mathbf{K}_{\text{nn}}\mathbf{v}_K^i, i = 1 \dots m$ can be reused, due to their computation in previous time steps. Since this is possible due to the cascaded use of the subspace projection extrapolation method, this modification is termed cascaded subspace projections extrapolation method.

5.2.2 Proper Orthogonal Decomposition for MRHS Problems

An alternative approach for computing well suited start vectors for the preconditioned conjugate gradient method is the method of proper orthogonal decomposition (POD). A well-known application of proper orthogonal decomposition is within the context of model order reduction, see e.g. [49, 50, 88]. At this, a larger model is transformed into a smaller substitute model containing significantly less degrees of freedom, but comprising most of the spectral information content of the original problem. The application of proper orthogonal decomposition for start vector generation was first proposed in [29, 79].

In a first step of the proper orthogonal decomposition method for the computation of an improved start vector in the $(m + 1)$ -th time step for solving (5.11) and (5.13), N_{POD} solution vectors of \mathbf{x} and \mathbf{a}_p from previous time steps are used to form the column vectors of the snapshot matrices \mathbf{X}_M and \mathbf{X}_K respectively. A singular value decomposition of each snapshot matrix \mathbf{X}_M and \mathbf{X}_K is evaluated [49, 50, 88]. The singular value decomposition (SVD) of \mathbf{X}_M results in

$$\mathbf{X}_M = \mathbf{U}_M \mathbf{\Sigma}_M \mathbf{V}_{\text{PM}}^\top. \quad (5.18)$$

Accordingly, the singular value decomposition of \mathbf{X}_K results in

$$\mathbf{X}_K = \mathbf{U}_K \mathbf{\Sigma}_K \mathbf{V}_{\text{PK}}^\top. \quad (5.19)$$

The matrices \mathbf{U}_M and \mathbf{V}_{PM} in equation (5.18) and the matrices \mathbf{U}_K and \mathbf{V}_{PK} in equation (5.19) are orthonormal matrices. The matrices $\mathbf{\Sigma}_M$ and $\mathbf{\Sigma}_K$ are diagonal matrices of the corresponding singular values stored on the main diagonal.

At this, the diagonal entries of Σ_M are the l_M singular values of \mathbf{X}_M ordered by magnitude, i.e., $\sigma_{Mi} \geq \sigma_{Mj}$ for $i < j$. The diagonal entries of Σ_K are the l_K singular values of \mathbf{X}_K ordered by magnitude, i.e., $\sigma_{Ki} \geq \sigma_{Kj}$ for $i < j$.

The classical SVD used in this thesis is based on the L_2 -norm. Alternatively, a weighted SVD, based e.g. on the energy norm, can be employed which takes into account information on the material parameters or edge lengths [35].

The numbers $l_{\hat{M}} < l_M$ and $l_{\hat{K}} < l_K$ are selected to keep most of the information from the largest singular values σ_{M1} and σ_{K1} , respectively, i.e.,

$$\frac{\sum_{i=1}^{l_{\hat{M}}} \sigma_{Mi}}{\sum_{i=1}^{l_M} \sigma_{Mi}} \stackrel{!}{\approx} 1 \quad (5.20)$$

and

$$\frac{\sum_{i=1}^{l_{\hat{K}}} \sigma_{Ki}}{\sum_{i=1}^{l_K} \sigma_{Ki}} \stackrel{!}{\approx} 1. \quad (5.21)$$

The relative information criterion in (5.20) and (5.21) usually is fulfilled if

$$\frac{\sigma_{Ml_{\hat{M}}}}{\sigma_{M1}} \leq \varepsilon_{\text{POD}}, \quad (5.22)$$

respectively

$$\frac{\sigma_{Kl_{\hat{K}}}}{\sigma_{K1}} \leq \varepsilon_{\text{POD}} \quad (5.23)$$

holds, where $\varepsilon_{\text{POD}} := 10^{-4}$ can be chosen for the threshold value ε_{POD} in practical applications [91].

The first $l_{\hat{M}}$ columns of \mathbf{U}_M are used to define an operator $\mathbf{U}_{\hat{M},\text{POD}} = [\mathbf{U}_{M:,1}, \dots, \mathbf{U}_{M:,l_{\hat{M}}}]$. Accordingly, the first $l_{\hat{K}}$ columns of \mathbf{U}_K are employed to define an operator $\mathbf{U}_{\hat{K},\text{POD}} = [\mathbf{U}_{K:,1}, \dots, \mathbf{U}_{K:,l_{\hat{K}}}]$.

An improved start vector $\mathbf{x}_{0,\text{POD}}$ for the preconditioned conjugate gradient method used to solve (5.11) for \mathbf{x} is obtained by evaluating

$$\mathbf{x}_{0,\text{POD}} := \mathbf{U}_{\hat{M},\text{POD}} \left[\mathbf{U}_{\hat{M},\text{POD}}^\top \mathbf{M}_{\text{cc}} \mathbf{U}_{\hat{M},\text{POD}} \right]^{-1} \mathbf{U}_{\hat{M},\text{POD}}^\top \mathbf{r}^{m+1}. \quad (5.24)$$

An improved start vector $\mathbf{a}_{p0,POD}$ for the preconditioned conjugate gradient method used to solve (5.12) for \mathbf{a}_p is obtained by evaluating

$$\mathbf{a}_{p0,POD} := \mathbf{U}_{\hat{K},POD} \left[\mathbf{U}_{\hat{K},POD}^\top \mathbf{K}_{nn} \mathbf{U}_{\hat{K},POD} \right]^{-1} \mathbf{U}_{\hat{K},POD}^\top \mathbf{K}_{cn}^\top \mathbf{a}_c^m. \quad (5.25)$$

In (5.24) and (5.24), the resulting matrices of the products $\left[\mathbf{U}_{\hat{M},POD}^\top \mathbf{M}_{cc} \mathbf{U}_{\hat{M},POD} \right]$ and $\left[\mathbf{U}_{\hat{K},POD}^\top \mathbf{K}_{nn} \mathbf{U}_{\hat{K},POD} \right]$ can be inverted efficiently using a direct method.

5.2.3 Numerical Validation: Improved Start Vectors

The numerical results stated in the following are published in [31, 29, 32].

The effect of computing improved start vectors for the preconditioned conjugate gradient method on the numerical simulation is investigated in this chapter. The TEAM 10 benchmark problem introduced in subchapter 5.1.1.2 is considered. The description of code, libraries and hardware used of subchapter 5.1.1 are also valid for the simulations described in the following.

In order to reduce the overall computation time, a time interval ranging from 0 to 120 ms is evaluated, instead of a time interval ranging from 0 to 150 ms, as selected in subchapter 5.1.1.2. Furthermore, in comparison to the simulations described in subchapter 5.1.1, the code for the implicit time integration is improved: The Newton-Raphson algorithm for linearization uses under-relaxation and lowered tolerances for faster convergence. Thus, the resulting simulation times for applying the implicit time integration scheme are smaller in this chapter than stated in subchapter 5.1.1.2. Two different meshes are considered in the following.

As a proof of the applicability of the code that is used for the numerical experiments, a very fine mesh resulting in about 700,000 degrees of freedom and the implicit Euler method for time integration with a time step size $\Delta t_{imp} = 1.0 \cdot 10^{-4}$ s is used to compare the simulation results with the measurement results published in [70]. The small time step length is used in order to reduce the number of Newton-Raphson iterations. As already explained in subchapter 5.1.1.2, the coding of the Newton-Raphson method does not involve a relaxation with an optimized relaxation value, which would allow a larger time step size without resulting in a large number of Newton-Raphson

iterations. The fine mesh resulting in about 700,000 degrees of freedom, is shown on the right side of Figure 5.14.

The results for the magnitude of the magnetic flux density evaluated along lines S1, S2, and S3, as described in subchapter 5.1.1.2, in comparison with the measurement results are shown in Figure 5.15. Here, it is obvious, that the resolution still is not fine enough to accurately resolve the skin effect, resulting in a deviation of the simulation results from the measurement results.

The resulting field plot of the magnetic flux density is shown on the right side of Figure 5.16. The simulation time for evaluating this problem with a time step size $\Delta t_{\text{imp}} = 1.0 \cdot 10^{-4} \text{ s}$ is 5.38 days on the hardware stated in subchapter 5.1.1.

Thus, in order to carry-out multiple simulations for both implicit and explicit time integrations with different parameters in an acceptable overall simulation time, a coarse mesh resulting in 29,532 degrees of freedom is considered. This coarse mesh is shown on the left side of Figure 5.14. In the following, the results obtained using the implicit Euler method for time integration are considered as state-of-the-art reference solutions against which the solutions obtained with the explicit Euler method for time integration are compared.

The coarse mesh yields a maximum stable time step size of $\Delta t = 1.2 \mu\text{s}$, for the explicit time integration scheme, which is also used as time step size for simulating.

For investigating the CSPE approach on evaluating a pseudo-inverse of \mathbf{K}_{nn} , the following parameters are varied in the simulations: The tolerance ε_{CG} for the stopping criterion of the preconditioned conjugate gradient method when evaluating the pseudo-inverse of the exterior curl-curl operator and a number N_{CG} of maximally acceptable conjugate gradient iterations, as explained in subchapter 5.2.1.

Values for the tolerance ε_{CG} considered are $\varepsilon_{\text{CG}} \in \{10^{-8}, 10^{-7}, 10^{-6}\}$. Values for the number N_{CG} of maximally acceptable conjugate gradient iterations are $N_{\text{CG}} \in \{1, 3, 5\}$.

For checking the resulting effect on the accuracy of the evaluated magnetic flux density and on the resulting simulation time, all combinations of ε_{CG} and N_{CG} are simulated. The field plot of the resulting magnetic flux density is shown on the left side of Figure 5.16.

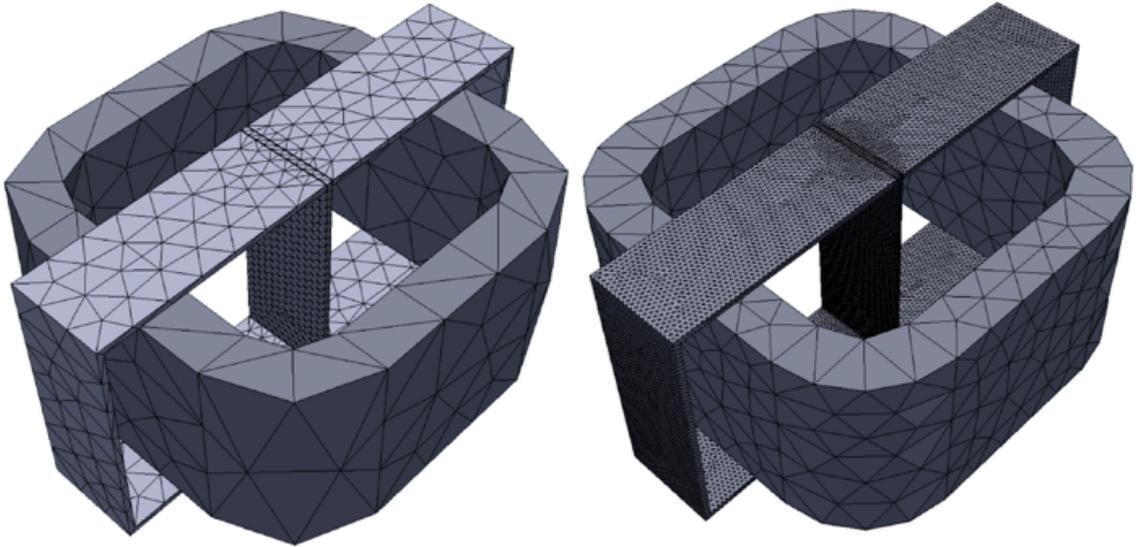


Figure 5.14: On the left side: Coarse mesh resulting in 29,532 degrees of freedom.
On the right side: Fine mesh resulting in about 700,000 degrees of freedom.

The resulting magnetic flux densities evaluated along lines S1, S2, and S3 are compared against the reference solution obtained using the implicit Euler method for time integration, for each tolerance ε_{CG} . This is shown in Figure 5.17. Even using the lowest tolerance $\varepsilon_{CG} = 10^{-6}$ does not result in a significant loss of accuracy.

Decreasing the tolerance ε_{CG} reduces the number of averagely required conjugate gradient iterations for convergence. These are also reduced by an increase of column vectors of the CSPE operator \mathbf{V}_K , as the information content for the calculation of the initial guess is increased.

The results are shown in Figures 5.18 and 5.19. The average number of conjugate gradient iterations resulting from applying the CSPE method is reduced by a factor of 4 to 12, compared to using the solution from the previous time step as start vector for the conjugate gradient method. This reduction holds for every chosen tolerance ε_{CG} .

For lower tolerances ε_{CG} , the effect of the number of column vectors in the operator

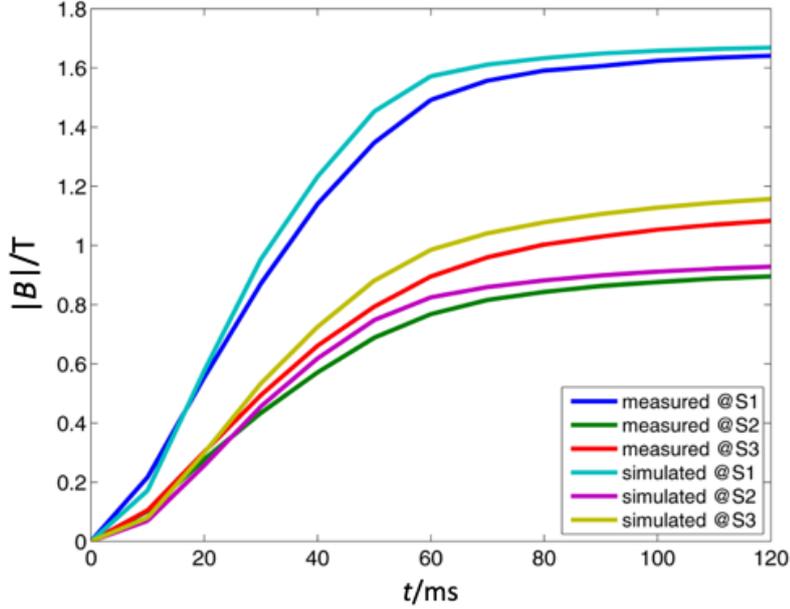


Figure 5.15: Comparison of simulation results evaluated on a fine mesh of 700,000 dofs and the measurement results published in [70] along lines S1, S2, and S3.

\mathbf{V}_K is less pronounced.

A rather small number of column vectors of less than 20 in the operator \mathbf{V}_K is sufficient as depicted in Figure 5.19.

Relaxing the tolerance ε_{CG} and applying the CSPE method for evaluating a pseudo-inverse of \mathbf{K}_{mn} reduces the simulation time as shown in Figure 5.20. Yet, a smaller number N_{CG} of maximally acceptable conjugate gradient iterations yields a longer simulation time. This is due to more computational effort in the CSPE method to gain so much information that the number of resulting required conjugate gradient iterations is decreased.

The application of the CSPE method reduces the required simulation time, compared to using the solution from the previous time step as start vector for the conjugate gradient method, by a factor of about 2.22. The simulations also show that it was not efficient to apply the CSPE method for evaluating the inverse of \mathbf{M}_{cc} . Here, the conjugate gradient method requires already so few iterations by using the previous time step's solution as start vector that the CSPE is an inefficient computational

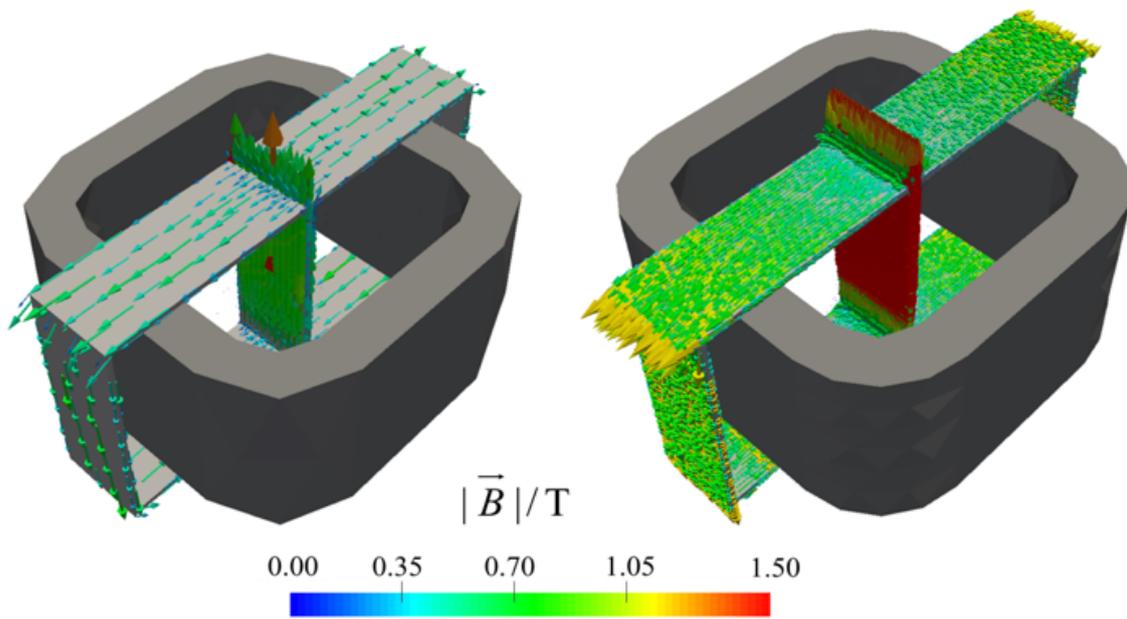


Figure 5.16: On the left side: Field plot of the magnetic flux density \vec{B} evaluated on the coarse mesh with 29,532 degrees of freedom.
On the right side: Field plot of the magnetic flux density \vec{B} evaluated on the fine mesh with about 700,000 degrees of freedom.

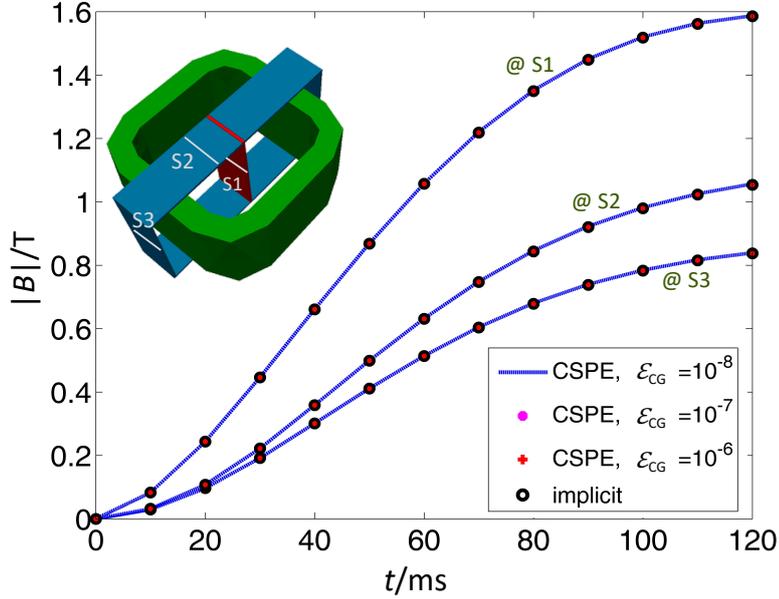


Figure 5.17: Simulation results obtained using the implicit Euler method and the explicit Euler method with varying CG tolerance evaluated along lines S1, S2, and S3.

overhead. Thus, in all simulations described in this subchapter, the solution from the previous time step is used as start vector when inverting \mathbf{M}_{cc} .

The simulation time of the reference method using the implicit Euler scheme for time integration is 2.35 h. Thus, further speed-up of at least a factor of 1.9 is required for becoming competitive to the standard method applying the implicit Euler scheme for the TEAM 10 benchmark problem discretized on the coarse mesh, as shown in Figure 5.14.

The results presented show that applying the CSPE method for computing improved start vectors for the preconditioned conjugate gradient method for evaluating a pseudo-inverse of \mathbf{K}_{nn} can significantly reduce the number of conjugate gradient iterations compared to using the previous time step's solution as start vector.

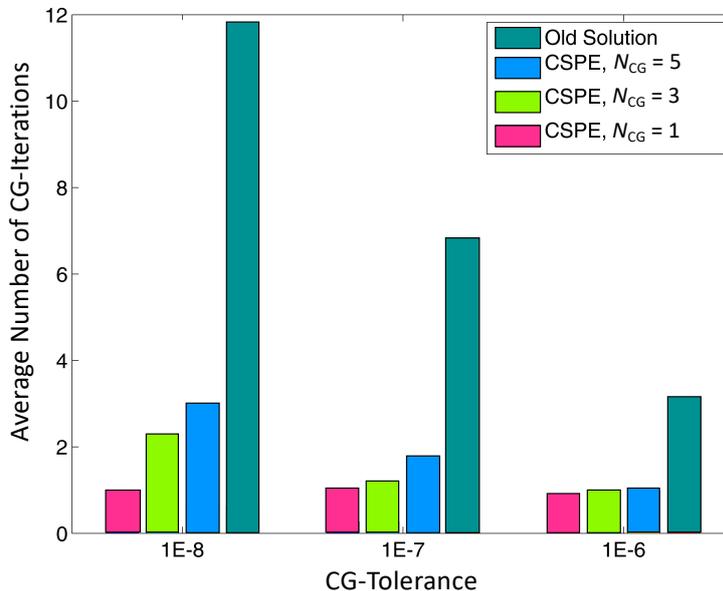


Figure 5.18: Averagely required number of PCG iterations when applying the CSPE method for computing improved start vectors for the PCG method.

Numerical simulations prove that this effect is also achieved by using POD for computing improved start vectors, as shown in Figure 5.21. In case of the CSPE, the spectral information content of the operator \mathbf{V}_K is increased by considering a larger number of column vectors of \mathbf{V}_K . It was shown, that less than 20 column operators in \mathbf{V}_K are sufficient. Thus, only small algebraic systems of equations have to be solved for computing an improved start vector, thus the computational effort is low resulting in reduced computation times.

This is different in case of using POD. According to (5.21), more than 99% of the spectral information is kept. Yet, the computational effort of the singular value decomposition and for computing the start vector according to (5.25) is higher than for applying the CSPE start vector generation method. Thus, although the number of conjugate gradient iterations is decreased significantly by applying POD, the simulation time is higher than when using the solution vector from the previous time step

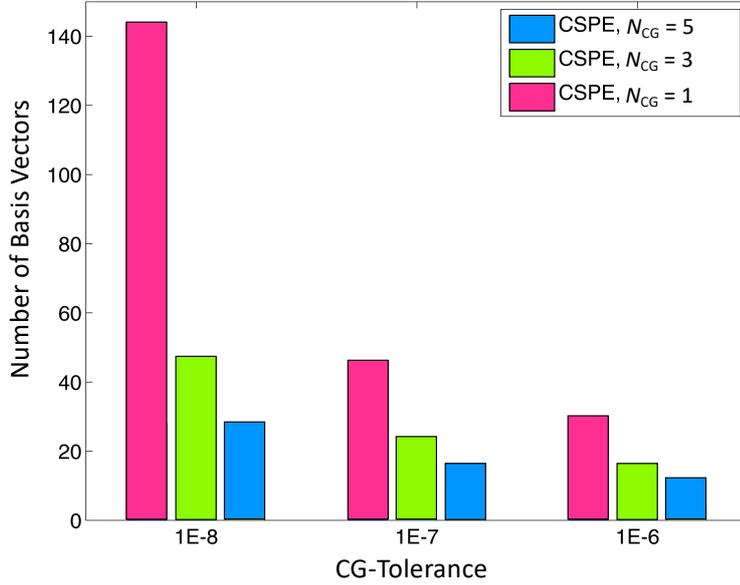


Figure 5.19: Maximum required number of column vectors in the operator \mathbf{V}_K of the CSPE method.

as start vector, as can be seen in Figure 5.22.

Yet, it should be noted that the application of POD can still be investigated and optimized further than presented in this thesis. The SVD is performed on solution vectors from an arbitrarily chosen number of previous time steps. It should be investigated whether a more suitable selection scheme for choosing these vectors would result in reduced computation effort, as e.g. published in [58].

With respect to applying POD for inverting the mass matrix \mathbf{M}_{cc} , the same holds as for applying CSPE: No significant improvement in performance compared to using the previous time step solution vector as start vector for the PCG method is observable.

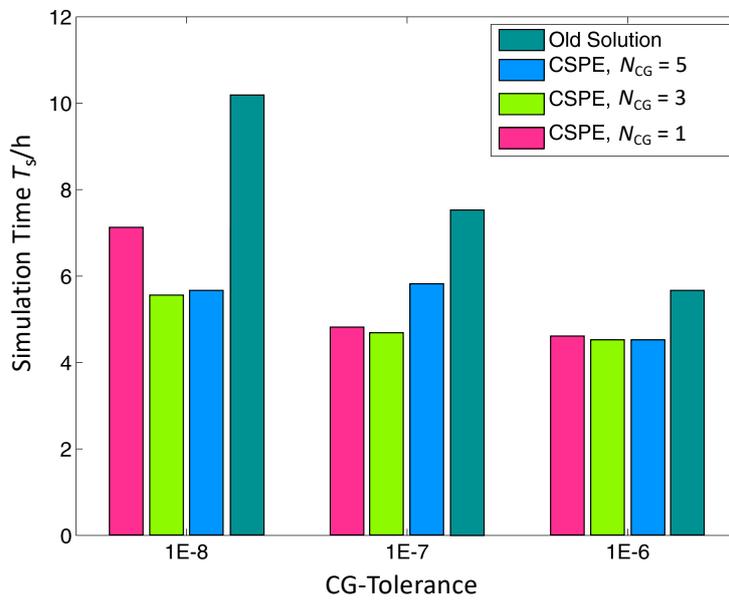


Figure 5.20: Resulting simulation times when applying the CSPE method for computing improved start vectors for the PCG method.

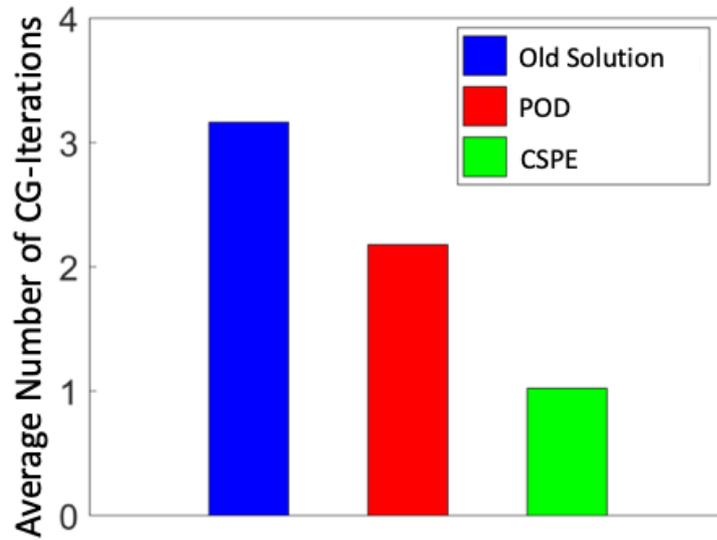


Figure 5.21: Averagely required number of PCG iterations using CSPE, or POD, or the old solution vector from the previous time step as start vector for the PCG method.

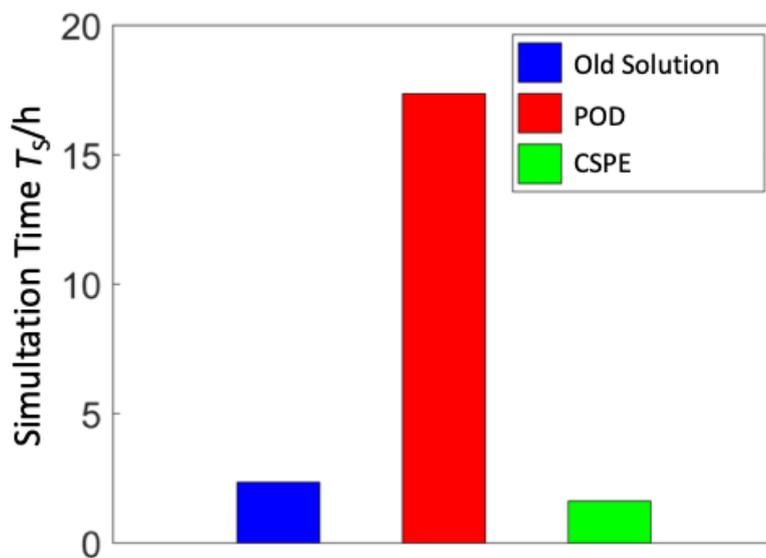


Figure 5.22: Simulation times using POD, CSPE, or the old solution vector from the previous time step as start vector for the PCG method.

5.3 Selective Matrix Update Bypass Strategy

As described in chapter 5, the numerical stability of the explicit Euler method, as of every explicit time integration scheme, is determined by the Courant-Friedrichs-Lewy criterion [19, 45]. The maximum stable time step size Δt for the time integration of equation (5.7) is given in equation (5.10). Here, the proportionality

$$\lambda_{\max} \left(\mathbf{M}_{\text{cc}}^{-1} (\mathbf{K}_{\text{cc}} (\mathbf{a}_c) - \mathbf{K}_S) \right) \propto \frac{1}{h^2 \kappa \mu} \quad (5.26)$$

holds for the maximum eigenvalue λ_{\max} , if the matrix \mathbf{M}_{cc} is regular, which is the case, if continuous edge element FEM ansatz functions are used. In equation (5.26), μ is the permeability, κ is the electrical conductivity, and h is mesh's smallest edge length. The maximum eigenvalue is computed using the Power Method [43].

Although the proportionality in (5.26) is useful as a qualitative information, it does not provide a sharp upper bound to the maximum eigenvalue λ_{\max} and cannot be used as computation method for λ_{\max} alternative to the Power Method. This is shown by numerical investigations.

Equation (5.26) shows, however, that meshes of fine resolution, i.e., meshes corresponding to small values of h , result in small maximum stable time steps. This is of practical relevance, especially in the simulation of real-world problems, where very fine-resolved meshes might be required, as e.g. in the air gap of three-dimensional electric machine models. Thus, values for the maximum stable time step size of the explicit Euler scheme for equation (5.5) often are in the micro- to nanosecond range.

With respect to the dynamics of excitation currents in magnetoquasistatic field problems, e.g. 50 Hz sinusoidal currents, applying such small time steps corresponds to a strong oversampling of the transient evolution of the problem.

For real-world problems, it can be expected, that only slight changes in the excitation current in directly succeeding time steps of explicit time integration schemes will result in only marginal changes in the entries of the vector \mathbf{a}_c in equation (5.7).

Following this assumption, a selective update scheme is proposed in [31, 24], where the matrix $\mathbf{K}_{\text{cc}}(\mathbf{a}_c)$ is not updated in every time step. Instead, the matrix $\mathbf{K}_{\text{cc}}(\mathbf{a}_c)$ is

only updated if

$$\frac{\|\mathbf{a}_c^m - \mathbf{a}_c^l\|}{\|\mathbf{a}_c^l\|} > tol, \quad l < m, \quad (5.27)$$

is valid, where changes in \mathbf{a}_c^l then only correspond to changes in the current excitation.

Here, tol is a chosen tolerance and $\|\cdot\|$ represents the L_2 -norm. In the m -th time step, the vector \mathbf{a}_c^m , i.e., the vector \mathbf{a}_c in the m -th time step, is computed. If it has changed from a vector \mathbf{a}_c^l , i.e., the vector \mathbf{a}_c computed in a previous l -th time step, more than a chosen tolerance tol , the matrix \mathbf{K}_{cc} is updated using \mathbf{a}_c^m . Here, the l -th time step is the time step in which the matrix \mathbf{K}_{cc} was last updated before.

The updating criterion (5.27) based on the L_2 -norm is used within the work of this thesis. However, other norms, as e.g. the magnetic energy norm, might be better suited, if another gauging is used.

5.3.1 Numerical Validation of the Selective Matrix Update Bypass Strategy

In this chapter, the effect of bypassing the update of the curl-curl matrix in conducting regions \mathbf{K}_{cc} is investigated by numerical simulations. The numerical results presented are originally published in [31, 24].

For the numerical test of this approach, the TEAM 10 benchmark problem introduced in subchapter 5.1.1.2 is considered. Again, the same code, libraries, and hardware explained in subchapter 5.1.1 are used for the simulations described in the following.

All simulations are based on the coarsely meshed problem resulting in a discrete MQS formulation with 29,532 degrees of freedom, as introduced in subchapter 5.1.1.2. Two different current excitations are considered: The current excitation introduced in subchapter 5.1.1.2 and shown in Figure 5.8, referenced hereafter as "switch-on current", and a ramped sinusoidal current with a 50 Hz frequency, hereafter referenced as "ramped sinusoidal current", as shown in Figure 5.23.

A time interval of 120 ms is simulated for the "switch-on current" and two periods of the "ramped sinusoidal current" are simulated. In both cases, the simulations start at

$t = 0$ s. The reason for introducing the "ramped sinusoidal current" is to check the applicability of the proposed bypassing strategy on problems which exhibit a more dynamic behavior than the original problem excited by the "switch-on current."

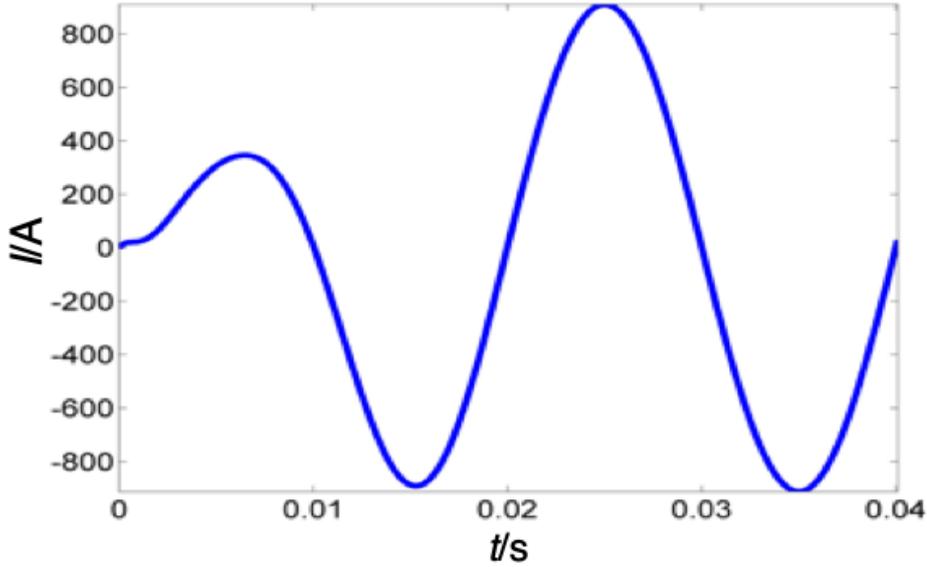


Figure 5.23: The first two periods of a 50 Hz ramped sinusoidal current, referred to as "ramped sinusoidal current".

The time step size used for time integration by the explicit Euler method is $\Delta t_{\text{CFL}} = \Delta t_{\text{exp}} = 1.2 \mu\text{s}$, resulting in 33,333 time steps in case of excitation by the "ramped sinusoidal current", and in 100,000 time steps in case of excitation by the "switch-on current".

The time step size $\Delta t_{\text{imp}} = 0.1$ s is used for reference simulations using the implicit Euler scheme for time integration. The CSPE method is used for computing improved start vectors for evaluating a pseudo-inverse of \mathbf{K}_{nn} with the PCG method in each time step. The tolerance for the preconditioned conjugate gradient method is set to 10^{-6} .

For the coarse mesh and first order edge elements, the resulting mass matrix \mathbf{M}_{cc} is merely of dimension $5,955 \times 5,955$ and can therefore be inverted directly using a parallelized GPU-accelerated LU-decomposition. For both excitation currents, several

simulations are executed with a varying parameter, i.e., the tolerance tol in equation (5.27). The values considered are $tol = 0.1\%$, 0.5% , 1.0% , 2.5% , 5.0% .

First, the magnetic flux density excited by the "switch-on current", computed by using the proposed method, is compared for all tolerances with the magnetic flux density computed by the reference simulation using the implicit Euler method and a time step size $\Delta t_{imp} = 0.1$ s.

The results for the magnetic flux density evaluated along the lines S1, S2, and S3, as described in subchapter 5.1.1.2, are shown in Figure 5.24.

The results obtained using the implicit Euler method and the results obtained using the explicit Euler method, with updating the matrix \mathbf{K}_{cc} in each time step, are in good agreement with the result obtained using the proposed method. This agreement is independent from the chosen tolerance tol .

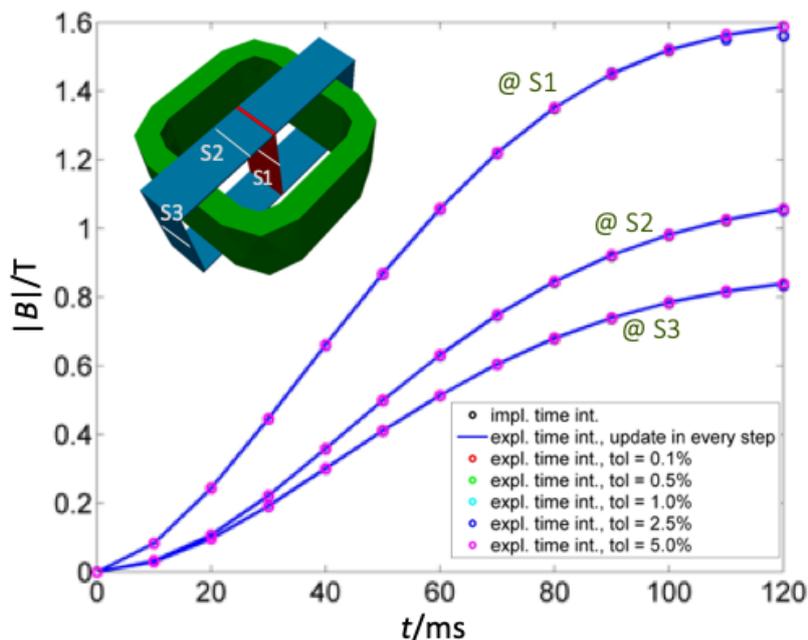


Figure 5.24: The first two periods of a 50 Hz ramped sinusoidal current, referred to as "ramped sinusoidal current".

The corresponding number of updates for the matrix \mathbf{K}_{cc} resulting for all chosen tolerances are shown in Figure 5.25. If the matrix \mathbf{K}_{cc} is updated in each time step, 100,000 updates are required during the entire simulation. Thus, even for a fine tolerance of

$tol = 0.1\%$ the number of updates that is still required is reduced to 7,000, which corresponds to a reduction of 93%. The corresponding simulation times, as well as

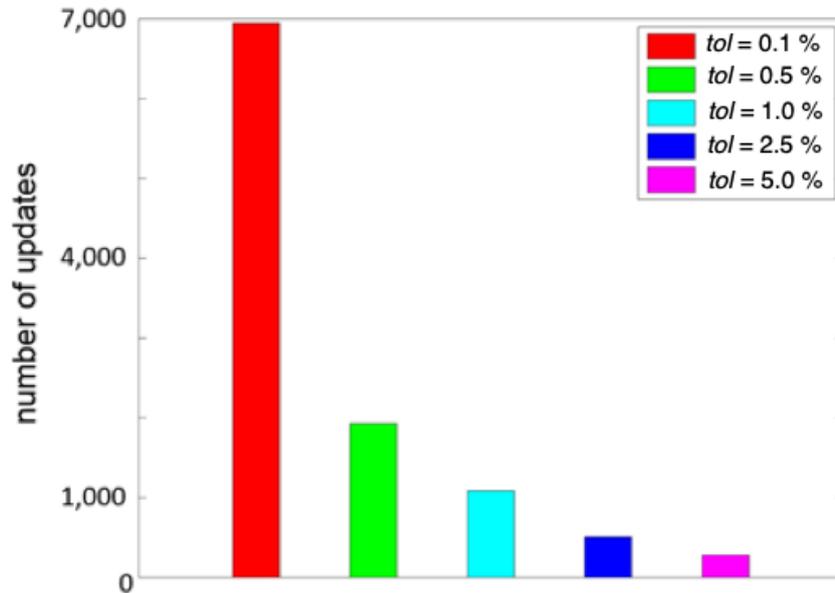


Figure 5.25: Number of updates for the matrix \mathbf{K}_{cc} and different tolerances, if the "switch-on current" is the excitation current.

the simulation times obtained when using the implicit Euler method for time integration, are shown in Figure 5.26. The proposed bypass strategy reduces the resulting simulation time below the time required for the reference simulation using the implicit time integration scheme for this test example.

However, the simulation times for different values for the tolerance tol do not deviate much from each other. This can be explained when considering the total simulation time of about 2 h, compared to the simulation time of about 4 h, if an update is performed in every time step. In case of $tol = 0.1\%$, only 7,000 updates within 100,000 time steps are required. This results in a 50% reduction of about 2 h in simulation time. If even less updates are performed, as in case of larger values for tol , the impact on the solution time is less pronounced.

In the following, the simulation results obtained using the "ramped sinusoidal current" as excitation current are investigated.

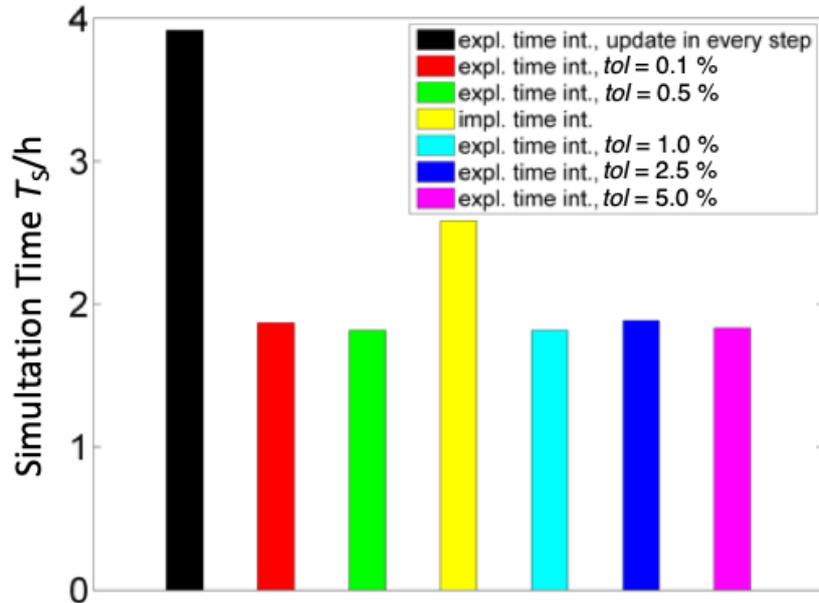


Figure 5.26: Simulation times T_S for the evaluation of the matrix \mathbf{K}_{cc} and all tolerances chosen for the "switch-on current" excitation.

The corresponding number of updates for the matrix \mathbf{K}_{cc} resulting for all chosen tolerances are shown in Figure 5.27. If the matrix \mathbf{K}_{cc} is updated in every time step, 33,333 updates are required during the entire simulation. As in case of the "switch-on current", even for a fine tolerance of $tol = 0.1\%$ the number of updates is reduced. Yet, in case of the fine tolerance $tol = 0.1\%$ the number of updates is about 50% of the maximum number of updates, compared to a reduction of 93% in the case of the "switch-on current" excitation. As expected, a more dynamic problem generally requires more updates.

The number of updates is further reduced for tolerances larger than $tol = 0.5\%$, but the effect is decreasing for increasing tolerances. The corresponding simulation times are shown in Figure 5.28. The simulation time is reduced for all chosen tolerances, but it does not change significantly for tolerance values larger than $tol = 0.5\%$. The simulation time is not significantly reduced further, as the remaining number of updates stays almost constant. Again, the remaining computational effort is not caused by the updating procedure, as is already shown for the results obtained by the "switch-on current".

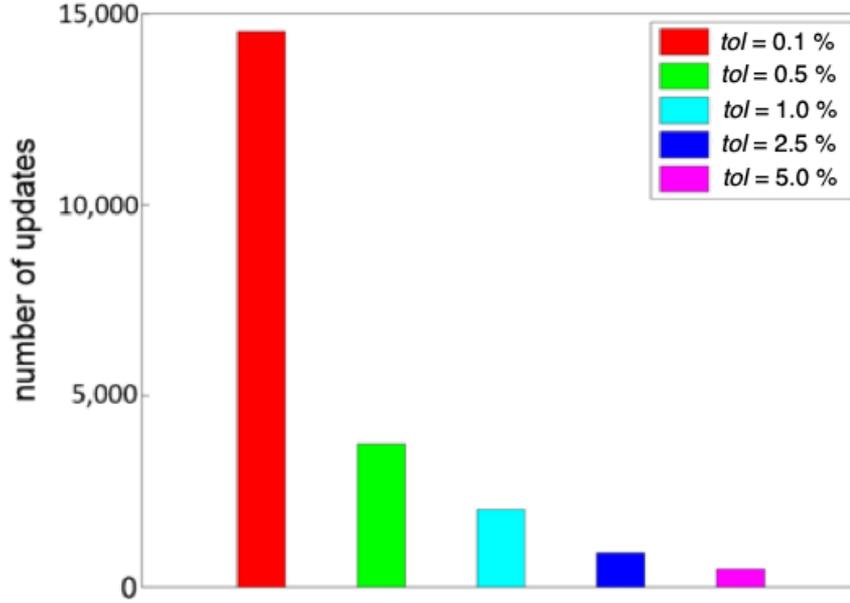


Figure 5.27: Resulting number of updates for the matrix \mathbf{K}_{cc} and all tolerances chosen, if the "ramped sinusoidal current" is the excitation current.

The loss of accuracy caused by bypassing the matrix update of $\mathbf{K}_{cc}(\mathbf{a}_c)$ is described by the error

$$error = \frac{|||\mathbf{B}_{sel}||_2 - ||\mathbf{B}_{evr}||_2|}{||\mathbf{B}_{evr}||_2}. \quad (5.28)$$

Here, the average magnetic flux density \mathbf{B}_{sel} is computed at position S1 using the selective update strategy and \mathbf{B}_{evr} is evaluated at position S1 updating the matrix $\mathbf{K}_{cc}(\mathbf{a}_c)$ in every time step. As expected, the error increases for larger tolerances. However, it remains sufficiently small even for a tolerance as large as $tol = 5.0\%$, for which the largest error is $tol = 0.9\%$, as plotted in Figure 5.29.

Now, the deviations between the average magnetic flux density evaluated using different time step sizes for the implicit Euler scheme are compared against the results obtained using the explicit Euler method for time integration with an update of the matrix \mathbf{K}_{cc} in every time step. The "ramped sinusoidal current" is used as excitation current. The error is described by

$$error = \frac{|||\mathbf{B}_{impl}||_2 - ||\mathbf{B}_{evr}||_2|}{||\mathbf{B}_{evr}||_2}. \quad (5.29)$$

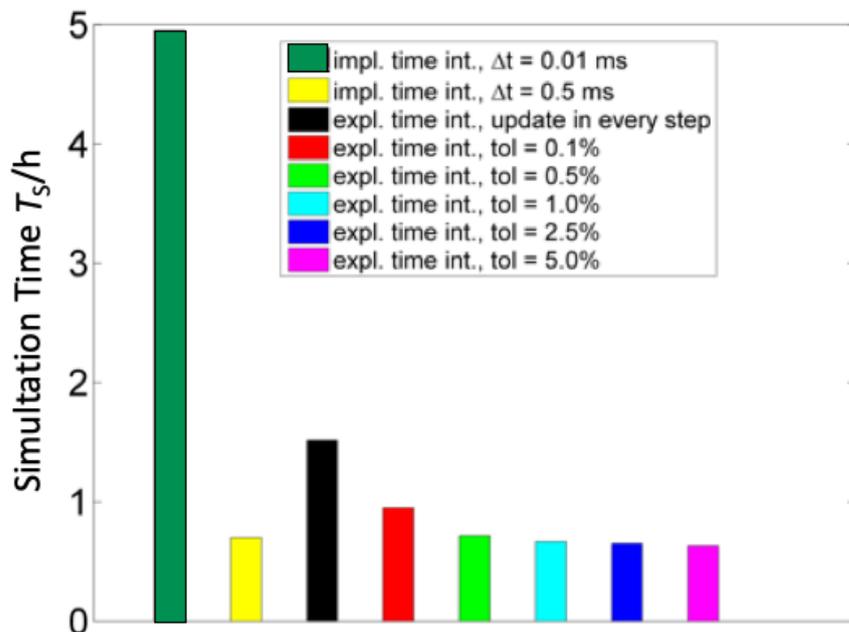


Figure 5.28: Simulation times T_s for the update strategy for the matrix \mathbf{K}_{cc} and all tolerances for the "ramped sinusoidal current" excitation current.

Here, \mathbf{B}_{impl} is the average magnetic flux density evaluated using the implicit Euler scheme. The results are shown in Figure 5.30.

From the plot shown in Figure 5.30, it is apparent that the results obtained with the implicit time integration method converge towards the results computed with the explicit time integration scheme for smaller time step sizes. In case of the implicit Euler method, the error is reduced for decreasing time step sizes, as it is first order convergent [92].

When using the implicit Euler scheme for time integration, a time step size of $\Delta t_{imp} = 0.01$ ms is needed for obtaining the same results as with the explicit Euler method. Thus, the explicit method is more accurate for this numerical test problem with less computational effort, since the resulting simulation time for the explicit method is about 5 times shorter than the simulation time of the implicit method with a time step size of $\Delta t_{imp} = 0.01$ ms for equally accurate results, as shown in Figure 5.28.

Yet, the spatial error is dominating. Thus, from an engineering point of view, applying the implicit Euler scheme with a time step size of $\Delta t_{imp} = 0.1$ ms may be considered

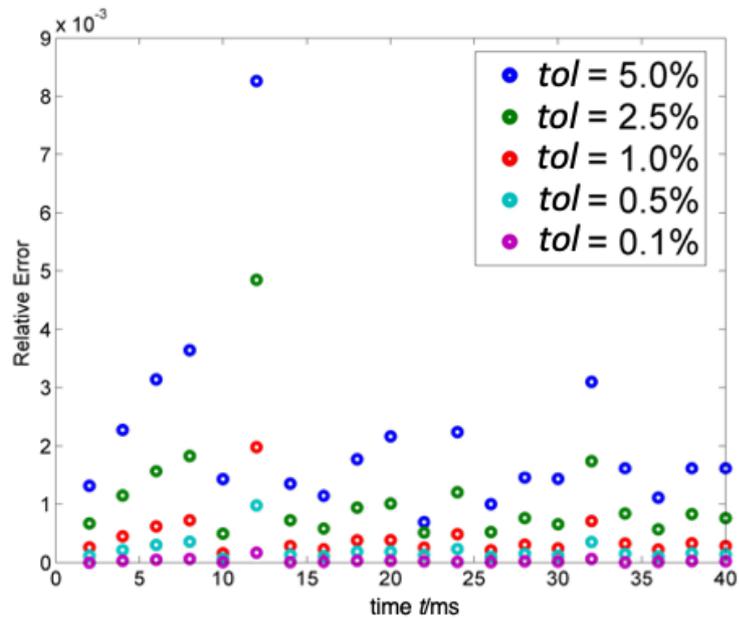


Figure 5.29: Relative error in the average flux density computed according to equation (5.28), comparing the selective update strategy vs. the full evaluation strategy.

to yield sufficiently accurate results. For this implicit time step size, the resulting simulation time is almost identical to the simulation time required by the proposed method.

Disadvantages of the proposed method with respect to order of convergence or stability are not observed by the numerical simulations. However, an analytic investigation on this is not performed.

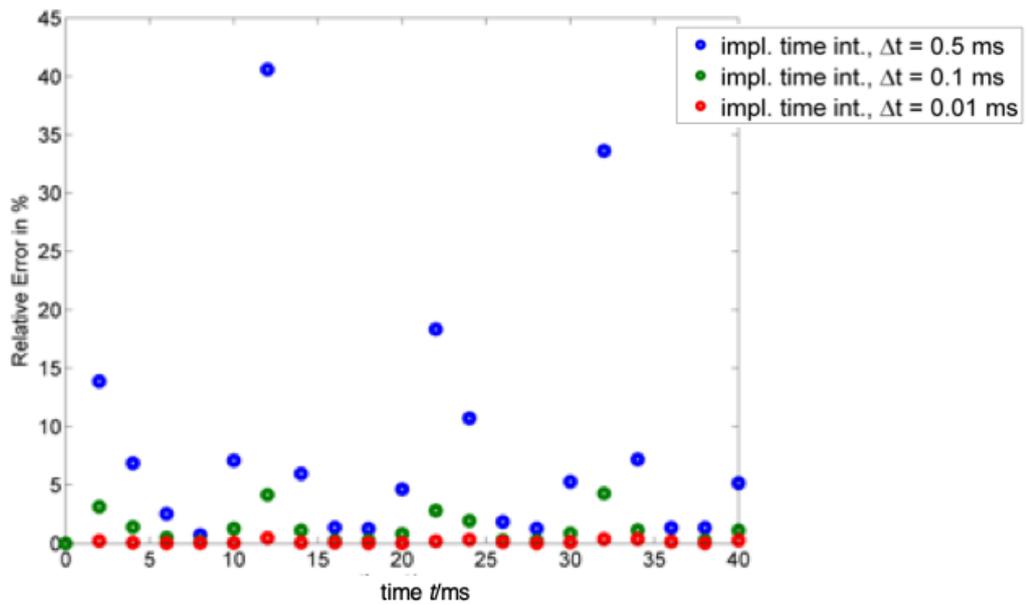


Figure 5.30: Deviation of the results computed for the average magnetic flux density along line S1 using the implicit Euler method and various time step sizes for time integration from the results obtained using explicit time integration and updating the matrix \mathbf{K}_{cc} in every time step.

5.4 GPU-Accelerated Inverse Computations

In previous chapters, a pseudo inverse of the singular curl-curl reluctivity matrix in nonconducting regions \mathbf{K}_{nn} is evaluated using the preconditioned conjugate gradient method. The inverse of the positive definite conductivity matrix in conducting regions \mathbf{M}_{cc} is not explicitly computed, but the corresponding algebraic systems of equations are solved using the preconditioned conjugate gradient method. In both cases, the preconditioned conjugate gradient method is initialized with improved start vectors using a multiple right-hand side technique, as e.g. the cascaded subspace projection extrapolation method or proper orthogonal decomposition. The advantage of this approach is, that it can be applied to arbitrary matrix dimensions of the matrices \mathbf{K}_{nn} and \mathbf{M}_{cc} .

If the dimension of the regular matrix \mathbf{M}_{cc} is sufficiently small to fit into the RAM of a GPU accelerator, the left application of the inverse within the time stepping scheme can be executed by forward-/backward substitutions that are part of a direct (LU-decomposition) method. At this, the LU-Decomposition of the matrix \mathbf{M}_{cc} is performed using GPU-acceleration. The LU-Decomposition of the matrix \mathbf{M}_{cc} is stored within the RAM of the GPU and can be reused in the subsequent forward-/backward substitution process for solving the multiple right-hand side problem (5.11).

This is a trivial approach to solve the equation system fast. However, due to the asymptotical complexity of the LU-Decomposition, it can be expected that this operation does not scale well with increasing matrix dimensions of \mathbf{M}_{cc} .

In this subsection, a computationally cheap approach for regularizing the singular matrix \mathbf{K}_{nn} is presented, without introducing a significant loss of accuracy. This was originally proposed in [27].

The presented approach's underlying idea is to regularize \mathbf{K}_{nn} by simply adding a small offset parameter $\epsilon \ll 1$ to each diagonal element of \mathbf{K}_{nn} . This approach is also used for regularization of singular matrices in the AMG method presented in [76, 77]. Thus, the regularized matrix $\mathbf{K}_{nn,reg}$ is inexpensively computed by evaluating

$$\mathbf{K}_{nn,reg} := \mathbf{K}_{nn} + \epsilon \mathbf{I}, \quad (5.30)$$

where \mathbf{I} is the identity matrix of the same dimension as \mathbf{K}_{nn} .

The inverse of the regularized matrix $\mathbf{K}_{nn,reg}$ is then computed explicitly employing the LU-Decomposition using GPU-acceleration. As in case of the matrix \mathbf{M}_{cc} , the LU-decomposition of the regularized matrix $\mathbf{K}_{nn,reg}$ is stored and reused for solving the corresponding multiple right-hand side problem (5.12).

The applicability of this approach is investigated empirically by choosing different values for ϵ and subsequent tests to show whether this results in a successful regularization and how much the result is affected by this non-physical modification of \mathbf{K}_{nn} .

5.4.1 Numerical Validation

The effect of adding a small regularizing offset value ϵ to the main diagonal of the matrix \mathbf{K}_{nn} and re-using GPU accelerated precomputed LU-decomposition of the so obtained matrix $\mathbf{K}_{nn,reg}$ and of the matrix \mathbf{M}_{cc} is investigated by numerical simulations. The numerical results presented in the following are originally published in [28, 27].

As in subchapters 5.2.3 and 5.3.1, the TEAM 10 benchmark problem introduced in sub-chapter 5.1.1.2 is used as test problem. The description of code, libraries and hardware used, stated in subchapter 5.1.1, is valid for the simulations described in the following. For the GPU-accelerated LU-decomposition, functions of the MAGMA library are used [107]. The MAGMA library provides functions and data structure for linear algebra on many-core systems, especially on GPUs and heterogeneous systems. Updates of the matrix \mathbf{K}_{cc} are bypassed, as described in subchapter 5.3.

Two different spatial discretizations are used for the numerical validations: First, the coarse mesh, already known from subchapters 5.1.1.2, 5.2.3, and 5.3.1 with a model of 29,532 degrees of freedom and a maximum stable time step size of $\Delta t_{CFL} = 1.2 \mu s$. A tolerance $tol = 2.5\%$ is used for the bypassing of updates of the matrix \mathbf{K}_{cc} . In addition, a finer mesh with a discrete model of 50,035 degrees of freedom and a maximum stable time step size of $\Delta t_{CFL} = 0.6 \mu s$ is used with a tolerance $tol = 1.0\%$ for the bypassing of updates of the matrix \mathbf{K}_{cc} .

This finer mesh is shown in Figure 5.31. This mesh results in a matrix \mathbf{M}_{cc} of dimension 15,639 x 15,639 and a matrix $\mathbf{K}_{nn,reg}$ of dimension 34,396 x 34,396. The LU-decomposition of larger matrices would exceed the available GPU-RAM of one

graphics processor, with 12 GB RAM, of an NVIDIA K80 GPU and would thus require the use of multiple GPUs.

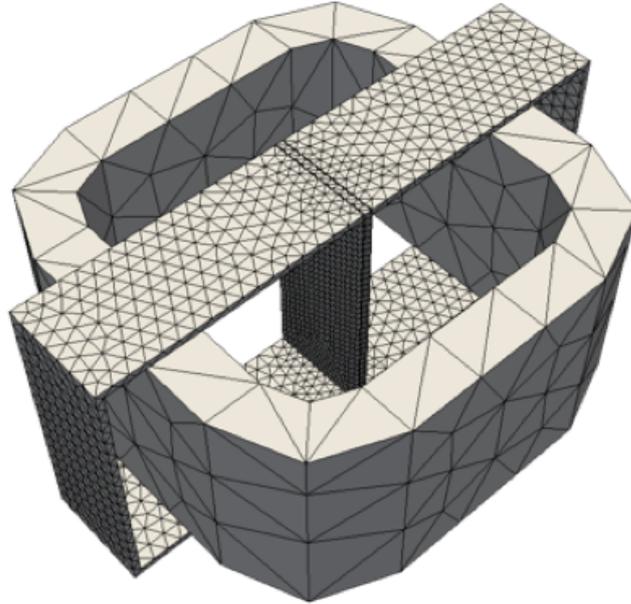


Figure 5.31: Mesh of the TEAM 10 problem resulting in 50,035 degrees of freedom.

A time interval ranging from 0 s to 120 ms of a current excitation described by a function $i_S(t) = (1 - \exp(-t/\tau))$ is simulated for both spatial discretizations, where $\tau = 0.05$ s is a time constant. Values for the regularization offset parameter ϵ are $\epsilon \in [10^{-9}, 10^{-12}, 10^{-15}]$. The results are presented in the following.

The magnitude of the average magnetic flux density is evaluated along line S3 for both meshes and all values of the regularization offset parameter ϵ . Simulations using the implicit Euler method for time integration, or the explicit Euler method with the CSPE-started PCG method for the evaluation of the pseudo-inverse of \mathbf{K}_{nn} , are used as reference simulations. The preconditioner used is an AMG-preconditioner from the Trilinos library [63, 51].

The results evaluated on the coarse mesh are shown in Figure 5.32; the results obtained on the fine mesh are shown in Figure 5.33. For both spatial discretizations, the results of all simulations are in good agreement. Thus, the application of a small offset value ϵ for regularizing \mathbf{K}_{nn} is a valid approach with respect to the accuracy of the evaluated

magnetic flux density. This makes the time integration scheme accessible for GPU-accelerated direct solution MRHS methods.

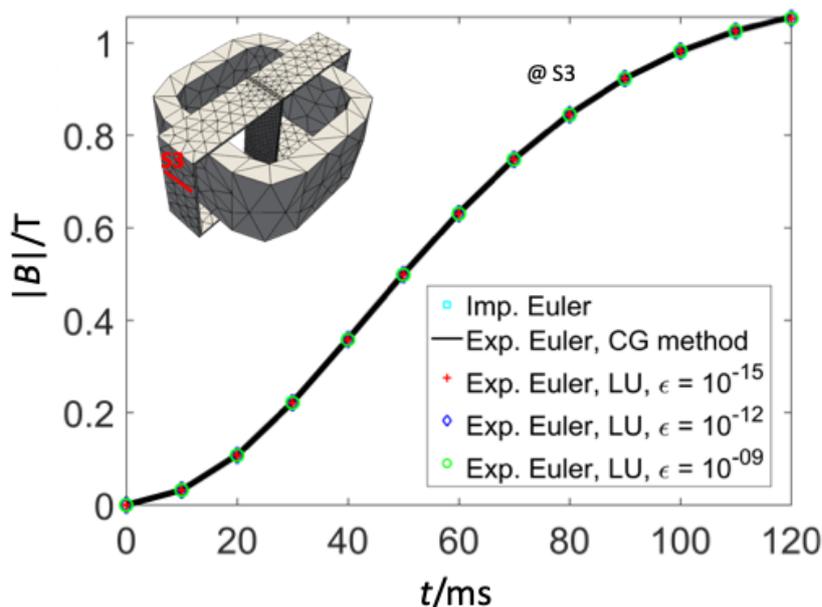


Figure 5.32: Results for the magnitude of average magnetic flux density on the coarse mesh evaluated along line S3.

The simulation times for all simulations carried-out on the coarse mesh are shown in Figure 5.34. As expected, the simulation time for all simulations using a GPU-accelerated LU-decomposition of the regularized matrix \mathbf{K}_{mn} are about equal independently from the exact value of the offset parameter ϵ .

These simulations are faster by 26 % compared to evaluating a pseudo-inverse of \mathbf{K}_{mn} using the AMG-preconditioned conjugate gradient method with improved start vectors computed by the CSPE method and faster by 52 % compared to using the implicit Euler scheme for time integration for the small test problems.

The simulation times for all simulations run on the finer mesh are shown in Figure 5.35. Again, the simulation time for all simulations using a GPU-accelerated LU-decomposition of the regularized matrix \mathbf{K}_{mn} are almost identical, independently from the exact value of the offset parameter ϵ , as expected. As in case of the simulations on the coarse mesh, these simulations are faster by 26 % compared to evaluating a pseudo-

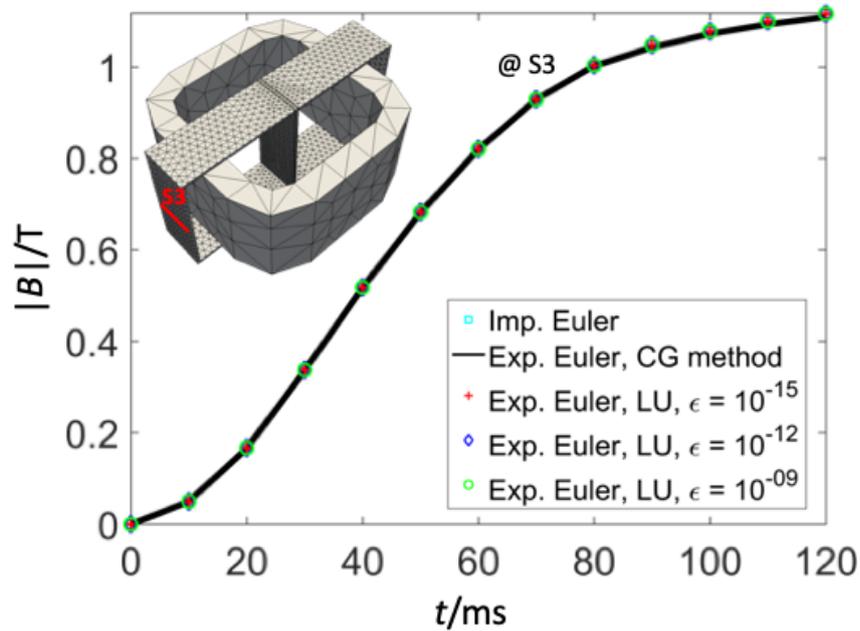


Figure 5.33: Results for the magnitude of average magnetic flux density on the fine mesh evaluated along line S3.

inverse of \mathbf{K}_{m} using the preconditioned conjugate gradient method with improved start vectors computed by the CSPE method.

However, they are significantly slower than the reference simulation using the implicit Euler scheme for time integration. Thus, the proposed method does not lead to a speed-up that can compensate for the small time step size.

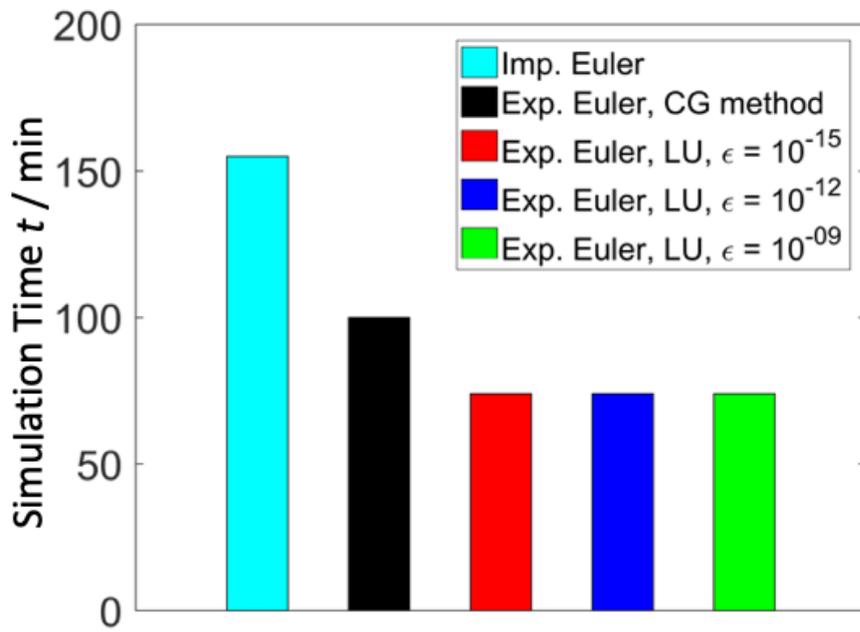


Figure 5.34: Simulation times on the coarse mesh.

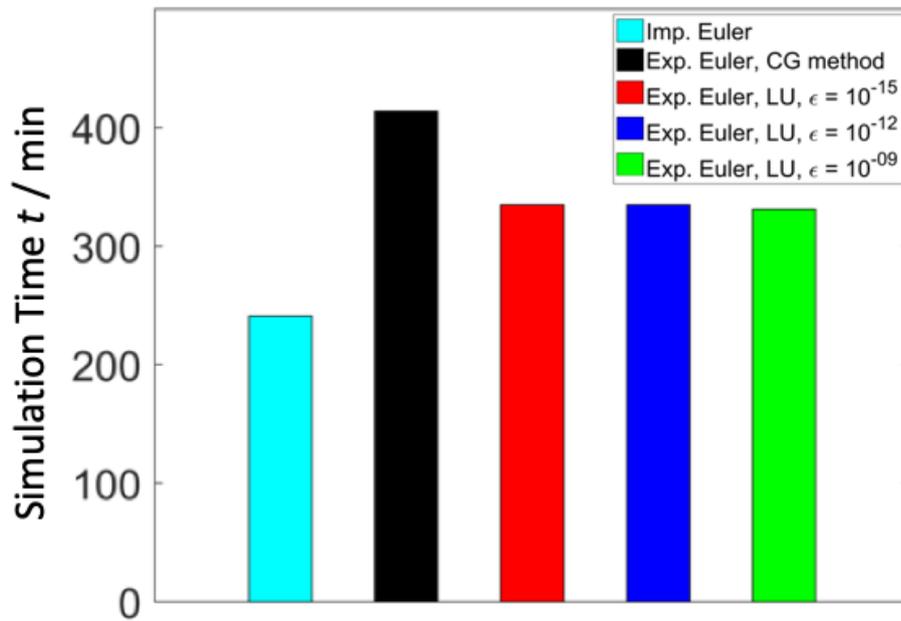


Figure 5.35: Simulation times on the finer mesh.

6 Parallel-In-Time Integration of Transient Electroquasistatic Field Problems Using Parareal

The Parareal algorithm is a parallel-in-time integration method for initial value problems, originally proposed in [62]. Usually, the spatial dimension is considered for parallelization, as the causality principle must be considered in temporal parallelization. However, parallelization in time is especially interesting, when employing massively parallel computers, and when the speed-up gained from spatial discretization saturates [73, 39]. The Parareal algorithm can be considered as both, a multiple shooting algorithm or as a multigrid method, with respect to time [41, 40].

In the following the application of the Parareal algorithm to electroquasistatic field problems is explained and numerically investigated.

6.1 Mathematical Formulation

As derived in chapter 2.1, the electroquasistatic approximation of Maxwell's equations is described by the partial differential equation

$$\nabla \cdot (\kappa(\phi) \nabla \phi) + \nabla \cdot \left(\varepsilon \nabla \frac{\partial \phi}{\partial t} \right) = 0. \quad (6.1)$$

Discretizing (6.1) in space using e.g. the finite element method yields a system of ordinary differential equations

$$\mathbf{M} \frac{d}{dt} \mathbf{x} + \mathbf{K}(\mathbf{x}) \mathbf{x} = \mathbf{b}, \quad (6.2)$$

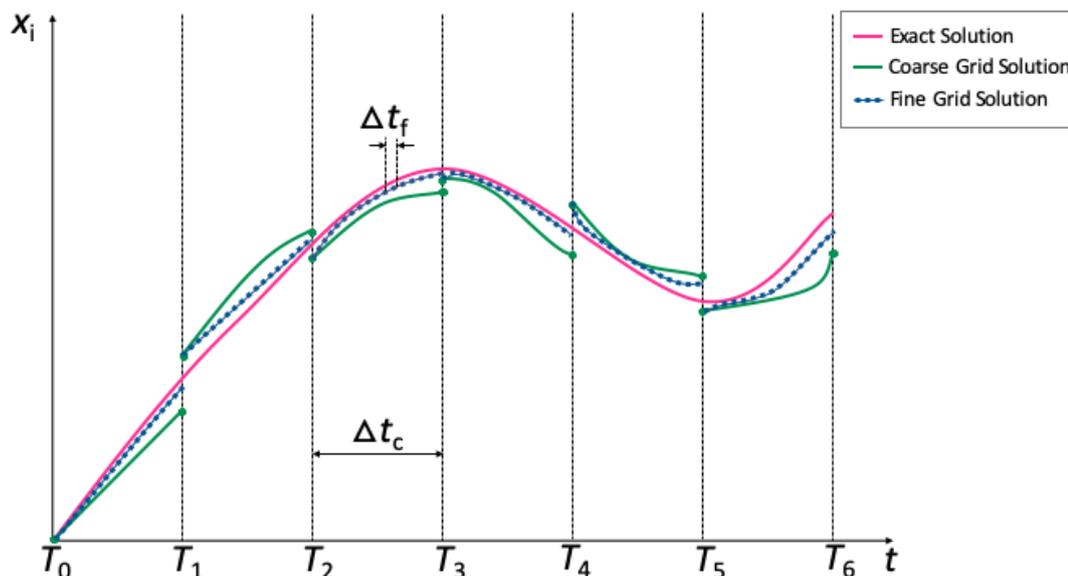


Figure 6.1: An exemplary plot of the solution on the coarse and the fine time grid is shown at one iteration of the Parareal algorithm. The coarse time grid consists of six time intervals with a time step size Δt_c . The solution for x on the coarse time grid is shown in green. The solution on the fine time grid with a time step size Δt_f is shown in each coarse time interval in blue. The exact solution, against which the Parareal algorithm's solution will converge, is shown in pink.

where \mathbf{M} is the mass matrix, \mathbf{x} is the vector of time-dependent electric scalar potentials, $\mathbf{K}(\mathbf{x})$ is the stiffness matrix depending nonlinearly on the electric potentials. The right-hand side \mathbf{b} contains the transient Dirichlet boundary conditions.

In the following, it is explained how the Parareal method is applied to integrate equation (6.2) in time following the presentation in [94], where the Parareal algorithm is applied to a magnetoquasistatic field problem.

The time interval to be simulated is split into N sub-intervals, where N is the number of parallel processes supported by the available computer hardware. These N time intervals constitute the starting points of the parallel time steps on the fine time grid. A propagator $G : I \times I \times \mathbb{R}^n \rightarrow \mathbb{R}^n$ propagates the solution along the coarse time grid with a coarse time step size Δt_c and another propagator $F : I \times I \times \mathbb{R}^n \rightarrow \mathbb{R}^n$

propagates the solution along the fine time grid with a time step size $\Delta t_f \ll \Delta t_c$. Here, the numerical implementation of G and F is either the implicit or explicit Euler method.

Thus, the solution $\bar{\mathbf{x}}_j$ on the coarse time grid at time T_j is obtained by evaluating $\bar{\mathbf{x}}_j = G(T_j, T_{j-1}, \mathbf{X}_{j-1})$, where \mathbf{X}_{j-1} is the initial value for solving the coarse grid problem in the time interval $[T_{j-1}, T_j]$. On the fine time grid at time T_j , the solution $\tilde{\mathbf{x}}_j$ is computed by $\tilde{\mathbf{x}}_j = F(T_j, T_{j-1}, \mathbf{X}_{j-1})$.

Algorithm 2 Parareal algorithm, as given in [94, 48]

Initialization: $\mathbf{X}_0^{(k)} \leftarrow \mathbf{x}_0, \bar{\mathbf{x}}_j^{(0)}, \tilde{\mathbf{x}}_j^{(0)} \leftarrow \mathbf{0}$ (for all j, k), set tolerance tol ;
 set Parareal iteration counter: $k \leftarrow 1$;
while $k \leq 2$ **or** $\max_j \|\mathbf{X}_j^{(k)} - \mathbf{X}_j^{(k-1)}\| > tol$ **do**
 for $j \leftarrow 1$ **to** N **do**
 coarse solution: $\bar{\mathbf{x}}_j^{(k)} \leftarrow G(T_j, T_{j-1}, \mathbf{X}_{j-1}^{(k)})$;
 post process: $\mathbf{X}_j^{(k)} \leftarrow \tilde{\mathbf{x}}_j^{(k-1)} + \bar{\mathbf{x}}_j^{(k)} - \bar{\mathbf{x}}_j^{(k-1)}$;
 end for
 parfor $j \leftarrow 1$ **to** N **do**
 fine solution: $\tilde{\mathbf{x}}_j^{(k)} \leftarrow F(T_j, T_{j-1}, \mathbf{X}_{j-1}^{(k)})$;
 end parfor
 increment counter: $k \leftarrow k + 1$;
end while

Time integration on the coarse and on the fine time grid is carried-out in an iterative process, which is explained in the following. At the time points T_j on the coarse time grid, continuity needs to be established. Therefore, matching conditions demanding equality of the solutions at this point in time obtained on the coarse and on the fine time grid are imposed. The equation system resulting from these matching conditions can be solved by Newton's method and yields a correction step described by

$$\mathbf{X}_j^{(k)} = \tilde{\mathbf{x}}_j^{(k-1)} + \bar{\mathbf{x}}_j^{(k)} - \bar{\mathbf{x}}_j^{(k-1)}, \quad (6.3)$$

as derived in [94]. Here, k is the iteration counter for the Parareal iterations, $\mathbf{X}_j^{(k)}$ is the corrected start value for time integration on the fine time grid in the time interval $[T_{j-1}, T_j]$ in the k -th iteration, $\tilde{\mathbf{x}}_j^{(k-1)}$ is the solution at time T_j evaluated on the fine time grid in the $(k-1)$ -th iteration, $\bar{\mathbf{x}}_j^{(k)}$ is the solution at time T_j evaluated on the

coarse time grid in the k -th iteration, and $\bar{\mathbf{x}}_j^{(k-1)}$ is the solution at time T_j evaluated on the coarse time grid in the $(k-1)$ -th iteration. This start-value correction step practically is a post-processing to the time integration on the coarse time grid.

The steps of the Parareal algorithm are listed in Algorithm 2. In a first step, all initializations are done, as e.g. setting the counter $k = 1$ and assigning an initial value \mathbf{x}_0 for time point T_0 . The initializations are only done once at the beginning of the computations.

In a second step, the time integration on the coarse time grid is carried-out. In a third step, the start values for time integration on the fine time grid are corrected according to equation (6.3). In a fourth step, each parallel processor is assigned a time interval $[T_{j-1}, T_j]$ and a corrected start value. The time integration on the fine time grid in all time intervals $[T_{j-1}, T_j]$ is carried out in parallel on N parallel processors. When all time integrations on the fine time grid are finished, the counter k is incremented by 1, i.e., $k \leftarrow k + 1$. In the last step of an iteration, the stopping criterion is evaluated. This is described by

$$\max_j \|\mathbf{X}_j^{(k)} - \mathbf{X}_j^{(k-1)}\| > tol, \quad (6.4)$$

as stated in [94], where tol is a prescribed tolerance. If the stopping criterion is fulfilled, the Parareal algorithm has converged towards a solution. If it is not fulfilled, another iteration follows.

An exemplary plot of obtained solutions in an iteration is shown in Figure 6.1. Here, the final, continuous solution and solutions obtained on the coarse and on the fine time grid are shown. These solutions converge towards the exact sequential solution after at maximum $k = N$ iterations of the Parareal algorithm [94].

6.2 Numerical Validation

A Matlab based inhouse FEM simulation code for two-dimensional electroquasistatic field problems is used with Matlab 2017b for the simulations [67]. The Matlab Parallel Toolbox is used for the parallelized computations on the fine time grid, i.e., the

parfor-loop in Algorithm 2. Neither the implicit nor the explicit Euler method are additionally parallelized. All simulations are run on a server with a 16-cores Intel Xeon E5-processor.

A two-dimensional model of an IEC 60099-4 surge arrester is used for numerical validation [55]. Two different triangular meshes are applied and both are spatially discretized using a nodal finite element ansatz and a Galerkin approach of the test functions. The meshes yield 23,577 and 51,528 degrees of freedom, respectively. For the tolerance in equation (6.4), a tolerance value $tol = 10^{-4}$ is used in all simulations.

Figure 6.2 shows the two-dimensional model geometry of the IEC 60099-4 surge arrester. The mesh resulting in 23,577 degrees of freedom is depicted in Figure 6.3.



Figure 6.2: Geometry of the two-dimensional model of an IEC 60099-4 surge arrester. The air region is colored in turquoise. The components of the surge arrester are colored in red, green, and orange. Areas, where the surge arrester pierces through the air plane are colored in white.

A sinusoidal excitation voltage with a frequency of 50 Hz and a peak value of 471 kV is applied to the surge arrester. A time interval ranging from 0 ms to 40 ms is simulated. The simulations are used as proof-of-concept for the applicability of the Parareal algorithm to electroquasistatic field problems. Therefore, in a first step, simulations with a linear field grading material are executed for both meshes. In a second step,

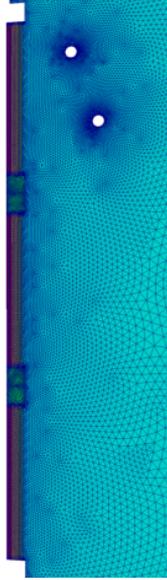


Figure 6.3: A triangular mesh resulting in 23,577 degrees of freedom is applied to the model geometry shown in Figure 6.2.

simulations with a nonlinear field grading material are performed for the mesh that results in a discrete model of 23,577 dofs.

For each mesh, four different simulations are executed:

- First, a simulation using only the implicit Euler method for time integration representing the general approach as reference simulation,
- the second is a simulation using only the explicit Euler method,
- the third simulation employs the Parareal algorithm with the implicit Euler method used for time integration on the coarse time grid and the implicit Euler method used for time integration on the fine time grid,
- the fourth simulation employs the Parareal algorithm with the implicit Euler method used for time integration on the coarse time grid and the explicit Euler method used for time integration on the fine time grid.

Simulations employing the Parareal algorithm and a stabilized explicit method for time integration on the coarse time grid and an explicit time integration scheme on the fine time grid are not executed. These simulations are of interest for future research. A

suitable stabilized explicit time integration method is e.g. the Runge-Kutta-Chebyshev method [103, 79, 78].

6.2.1 Results of Simulations with a Linear Field Grading Material

An overview of the simulations and the chosen time step sizes for both meshes are shown in Table 6.1 and in Table 6.2.

Name	Time Integration Method	Time Step Size
Imp. Euler	Implicit Euler	0.1 ms
Exp. Euler	Explicit Euler	0.1 ms

Table 6.1: Time step lengths for simulations with linear field grading material using the implicit and the explicit Euler method for the sequential time integration.

Name	Time Int. Coarse Grid	Time Int. Fine Grid	Δt_c	Δt_f
ImIm-Parareal	Implicit Euler	Implicit Euler	2.5 ms	0.1 ms
ImEx-Parareal	Implicit Euler	Explicit Euler	2.5 ms	0.1 ms

Table 6.2: Time step lengths for simulations with linear field grading material using the Parareal algorithm.

The evaluated results for the electric potential and for the voltage stress of the problem with 23,577 degrees of freedom are shown in Figure 6.4 and in Figure 6.6. The evaluated results for the electric potential and for the voltage stress of the problem with 51,528 degrees of freedom are shown in Figure 6.5 and in Figure 6.7.

The results of the reference simulations and of the simulations employing the Parareal algorithm for time integration are in good agreement, which validates the applicability of the Parareal algorithm to the problems considered.

The resulting simulation times for the problem with 23,577 degrees of freedom are depicted in Figure 6.8. The resulting simulation times for the problem with 51,528 degrees of freedom are depicted in Figure 6.9.

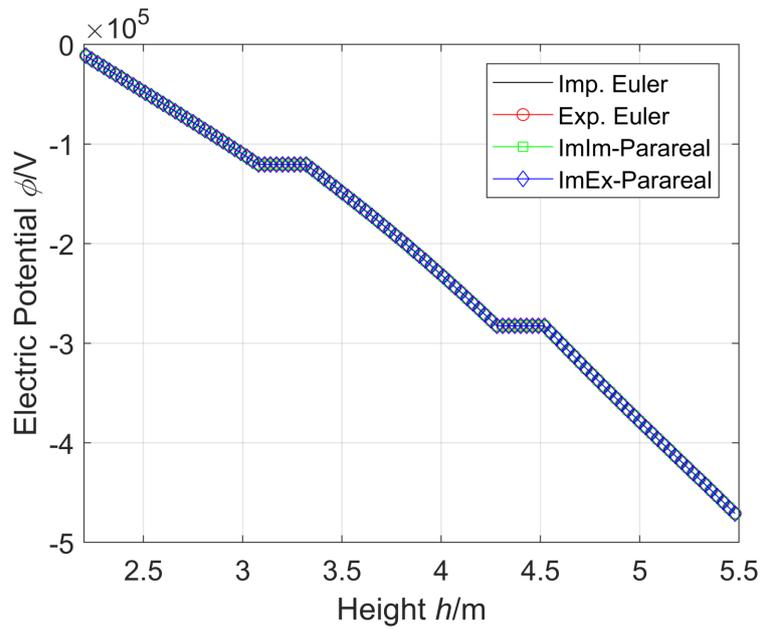


Figure 6.4: Results for the electric potential of the mesh resulting in 23,577 degrees of freedom with linear field grading material.

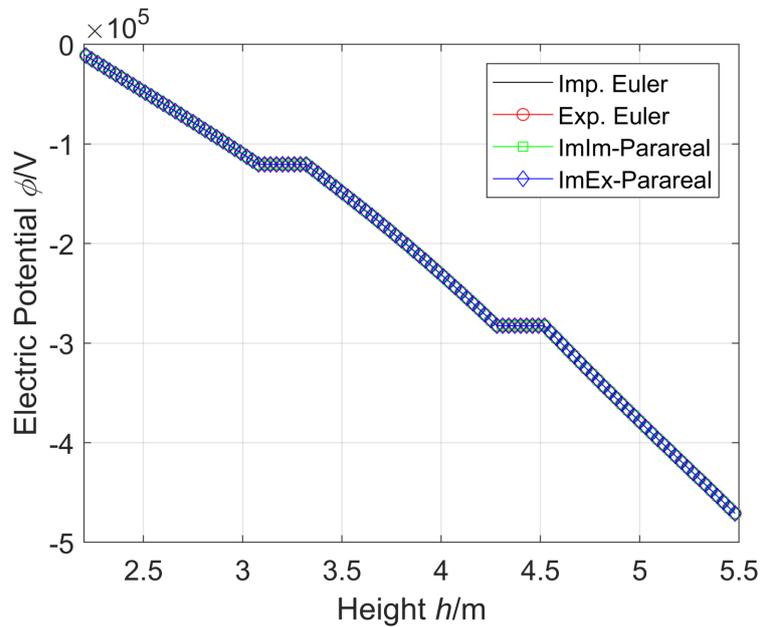


Figure 6.5: Results for the electric potential of the mesh resulting in 51,528 degrees of freedom with linear field grading material.

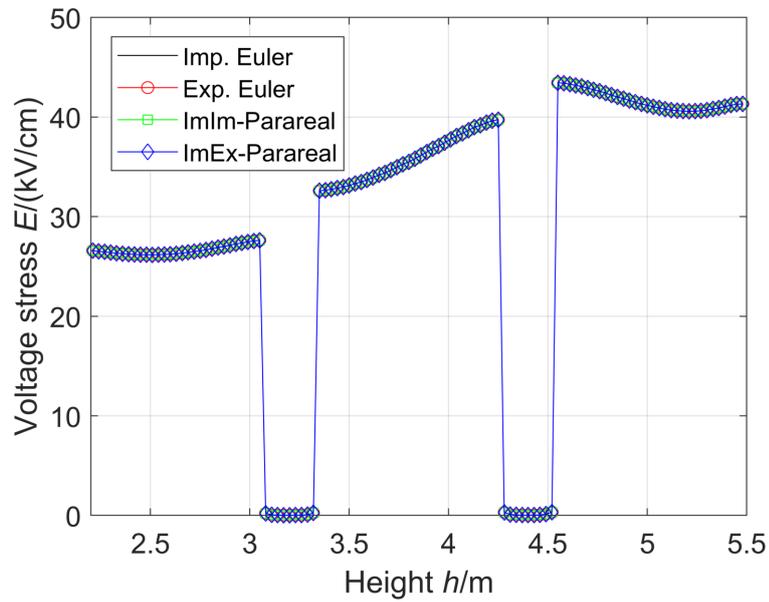


Figure 6.6: Results for the voltage stress of the mesh resulting in 23,577 degrees of freedom with linear field grading material.

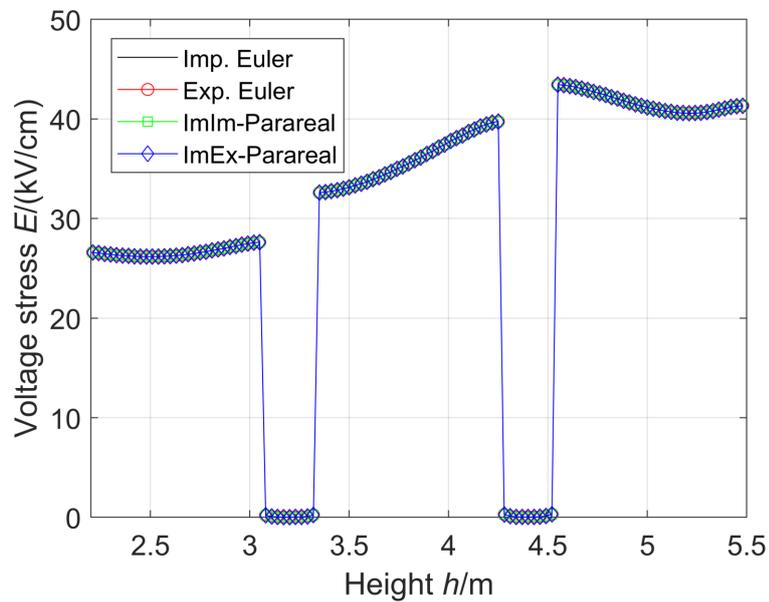


Figure 6.7: Results for the voltage stress of the mesh resulting in 51,528 degrees of freedom with linear field grading material.

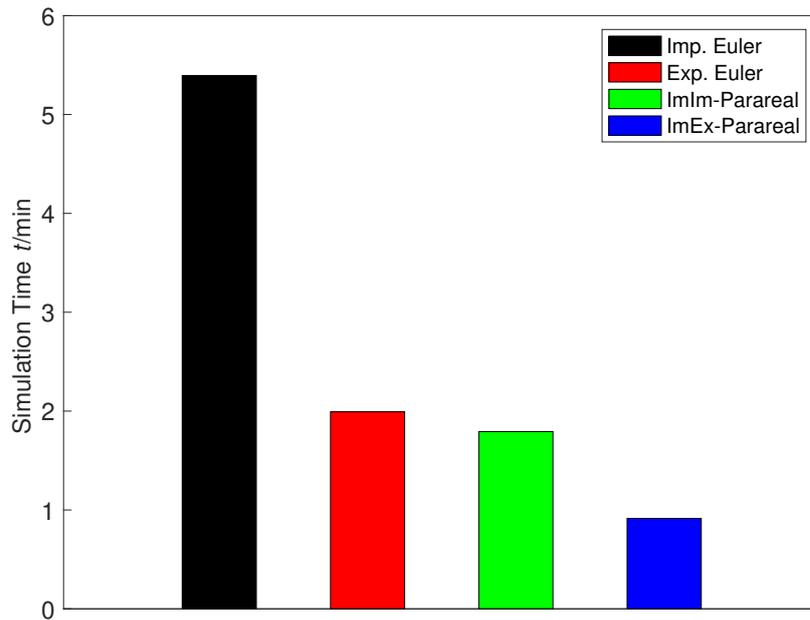


Figure 6.8: Real simulation times of the problem with 23,577 degrees of freedom and with linear field grading material.

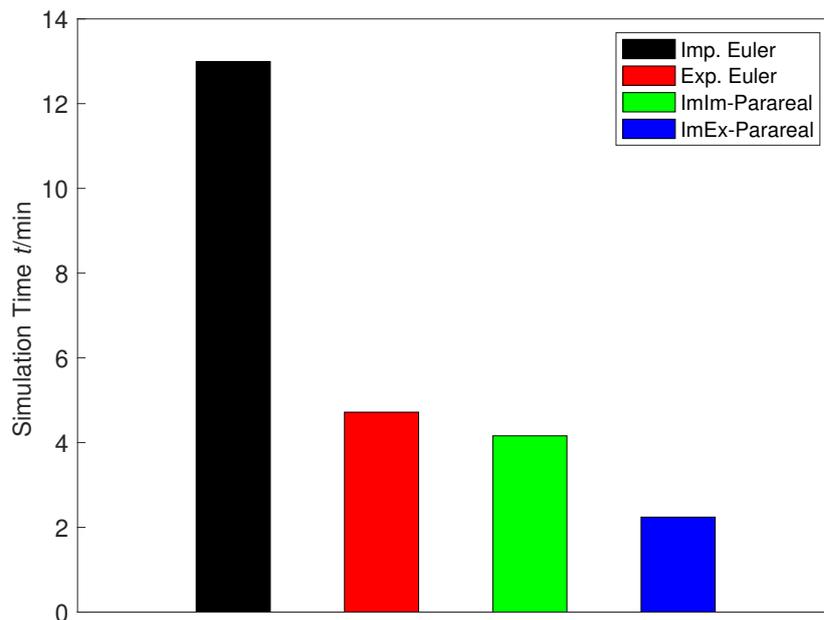


Figure 6.9: Real simulation times of the problem with 51,528 degrees of freedom and with linear field grading material.

As expected, the time for the simulations using the Parareal algorithm are significantly faster than the reference simulations using the implicit Euler method for time integra-

tion. The simulation time resulting from using the Parareal algorithm with the implicit Euler method on the coarse time grid and the implicit Euler method on the fine time grid is only slightly faster than the reference simulation employing the explicit Euler method.

The simulation time resulting from applying the Parareal algorithm with the implicit Euler method on the coarse time grid and the explicit Euler method on the fine time grid requires only about half the simulation time of the reference simulation using the explicit Euler method. Thus, at least for the problems considered, the Parareal algorithm, especially when using the explicit Euler method on the fine time grid, is shown to be a competitive approach with respect to simulation time.

6.2.2 Results of Simulations with a Nonlinear Field Grading Material

In the following, the simulation results obtained using a nonlinear field grading material for the mesh that yields a discrete model with 23,577 dofs, are presented. The conductivity curve of the nonlinear field grading material used is shown in Fig. 6.10.

Again, four different simulations are executed using the following schemes for time integration:

- the implicit Euler method,
- the explicit Euler method,
- the Parareal algorithm using the implicit Euler method on the coarse time grid, and the implicit Euler method on the fine time grid,
- the Parareal algorithm using the implicit Euler method on the coarse time grid, and the explicit Euler method on the fine time grid,

The time step sizes employed are presented in Table 6.3 and in Table 6.4

It should be noted, that a very small implicit time step size is used for the reference simulation employing the implicit Euler scheme. The reason for this, is that the coding of the Newton-Raphson method does not involve a relaxation scheme. For larger time

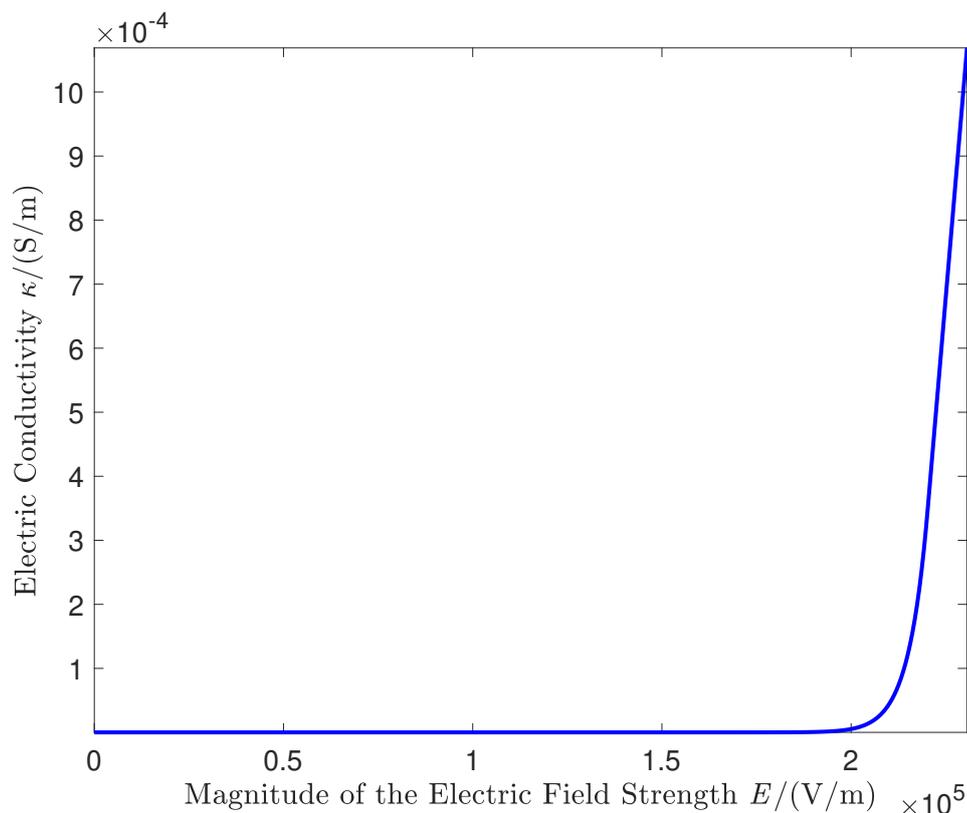


Figure 6.10: Conductivity curve of the nonlinear field grading material of the surge arrester used.

step size of the implicit Euler method, significantly more Newton-Raphson iterations are required than for small time step lengths. This also causes the long simulation time of the variant where the Parareal algorithm with the implicit Euler method for time integration on the coarse and on the fine time grid (ImIm-Parareal) is used, as shown in Figure 6.11. The time step size on the coarse time grid $\Delta t_c = 2$ ms results in many time consuming Newton-Raphson iterations. For easier comparability of the simulation times of the other three simulations, Figure 6.12 shows all simulation times except for the simulation time of the ImIm-Parareal simulation. Amongst the simulations executed, ImEx-Parareal requires the shortest simulation time. However, these simulations only serve as proof-of-concept for the applicability of the Parareal algorithm to this test problems. Using a different code, e.g. with an optimal relaxation for the Newton-Raphson method, or other optimizations with respect to simulation time, can result in completely different simulation times.

Name	Time Integration Method	Time Step Size
Imp. Euler	Implicit Euler	1 μ s
Exp. Euler	Explicit Euler	1 μ s

Table 6.3: Time step lengths for simulations with a nonlinear field grading material using the implicit and the explicit Euler method for the sequential time integration.

Name	Time Int. Coarse Grid	Time Int. Fine Grid	Δt_c	Δt_f
ImIm-Parareal	Implicit Euler	Implicit Euler	2 ms	1 μ s
ImEx-Parareal	Implicit Euler	Explicit Euler	10 μ s	1 μ s

Table 6.4: Time step lengths for simulations with a nonlinear field grading material using the Parareal algorithm.

The results for the electric potential and for the voltage stress are shown in Figure 6.13 and in Figure 6.14. Contrary to the simulation results obtained with a linear field grading material, the results computed with the nonlinear field grading material differ for the time integration schemes used. However, the agreement is still sufficiently good for most practical purposes. The reason for the deviations is not investigated further within this work. A possible assumption is, that the results evaluated by using the explicit Euler method for time integration are more accurate than those obtained with the implicit Euler scheme, as linearization is avoided. The reason for the deviation of the results computed using the ImIm-Parareal algorithm requires further investigations.

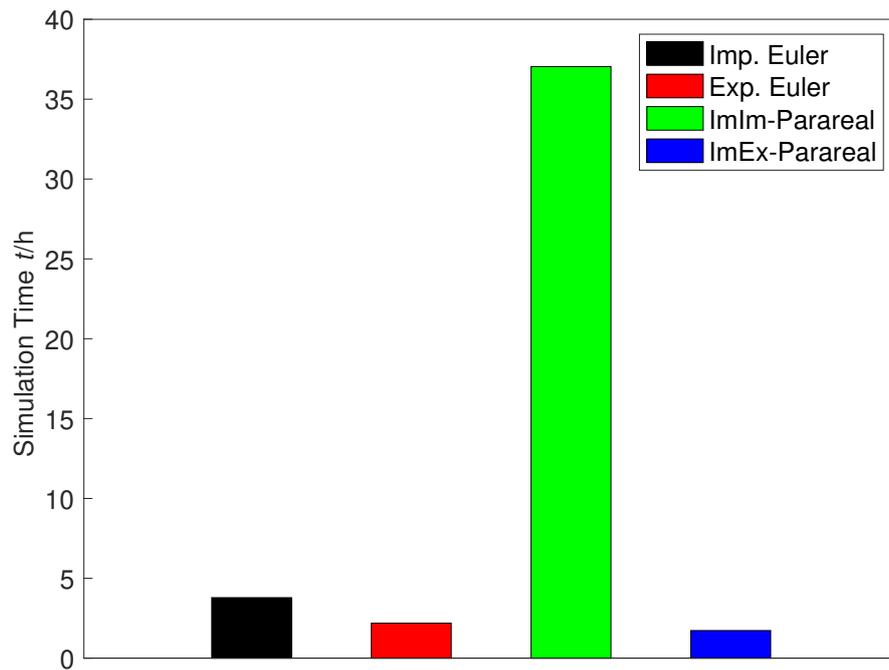


Figure 6.11: Real simulation times for the mesh resulting in 23,577 degrees of freedom with a nonlinear field grading material.

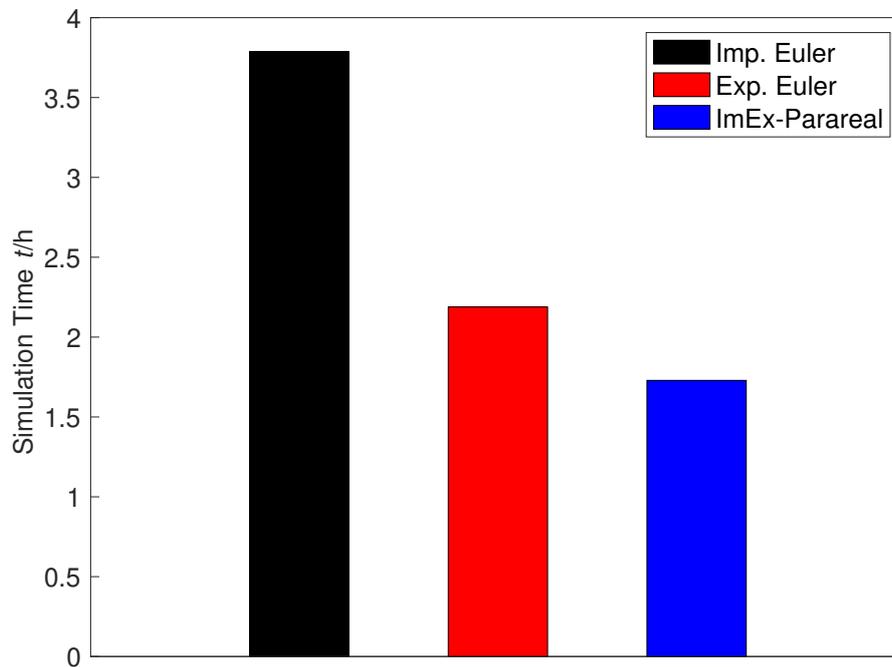


Figure 6.12: Real simulation times for the mesh resulting in 23,577 degrees of freedom with a nonlinear field grading material, without the simulation time without ImIm-Parareal.

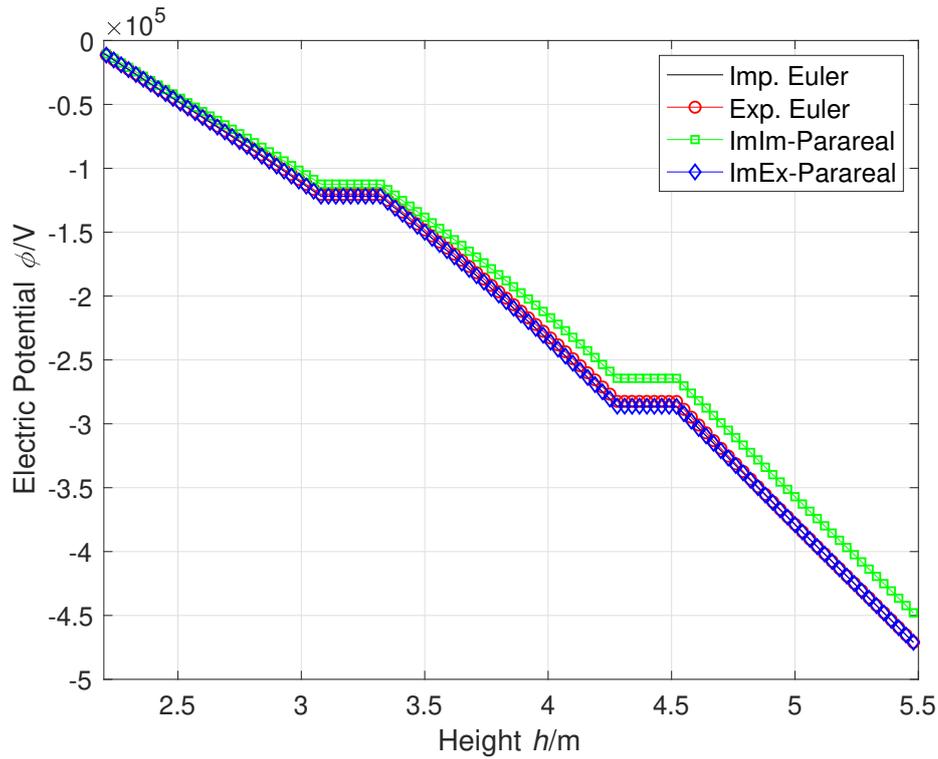


Figure 6.13: Results for the electric potential of the mesh resulting in 23,577 degrees of freedom with a nonlinear field grading material.

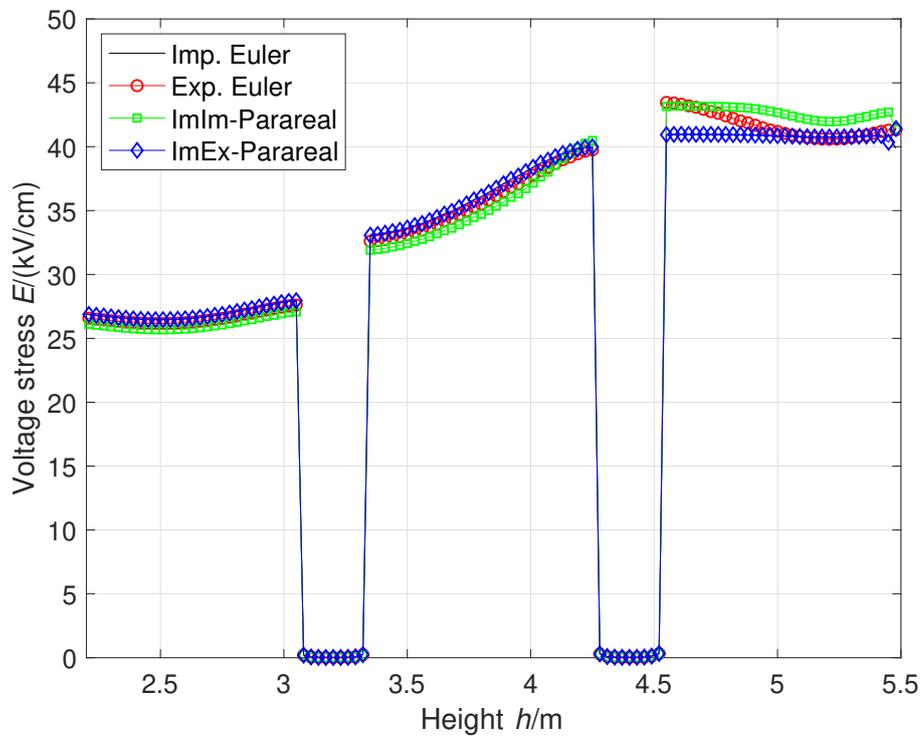


Figure 6.14: Results for the voltage stress of the mesh resulting in 23,577 degrees of freedom with a nonlinear field grading material.

7 Conclusion and Outlook

The main part of this thesis reports on the (semi-) explicit time integration of magneto-quasistatic field problems. In order to allow the application of explicit time integration schemes, the infinitely stiff differential algebraic equation of index 1 resulting from the spatial discretization of the partial differential equation of the magnetic vector potential formulation was transformed into a system of finitely stiff ordinary differential equations by application of a generalized Schur complement. For this, the degrees of freedom were separately allocated depending on their allocation in either conducting or nonconducting subregions of the computational domain.

The application of the Schur complement and the computations required in each time step of the explicit Euler method, still require the solution of algebraic linear systems of equations repeatedly, thus the term semi-explicit time integration. Equation systems that need to be solved involve the inverse of the regular mass matrix and the evaluation of a pseudo-inverse of the singular curl-curl operator in nonconducting (air) regions.

The application of the explicit Euler method avoids the need for solving large nonlinear systems of equations including computationally expensive linearization methods, such as the Newton-Raphson method including the ill-conditioned numerical evaluation of a Jacobian matrix. The main disadvantage of explicit time integration schemes in general is a commonly small maximum stable time step size according to the Courant-Friedrich-Levy criterion. In many realistic model configurations this results in a strong oversampling of the transient evolution due to stability constraints. Thus, the focus of this thesis was to explore various methods to speed-up the computations and compensate for the small time step size.

The first approach presented for acceleration involves the application of multiple-right hand side methods for the mass matrix inverse and the pseudo-inverse of the singular curl-curl operator in nonconducting regions. For this, the cascaded subspace

extrapolation method and the proper orthogonal decomposition method for computing improved start vectors for the preconditioned conjugate gradient method were investigated. Both methods successfully reduced the resulting number of conjugate gradient iterations, but only the cascaded subspace extrapolation method resulted in a reduction of simulation time, due to its low computational effort. Especially the proper orthogonal decomposition method was not sufficiently intensively investigated within the work of this thesis. A more suitable selection algorithm for input vectors for the singular value decomposition would certainly reduce the computational effort and would improve the competitiveness of the proper orthogonal decomposition methods. This is an important aspect of future work.

Another approach for a reduction of the simulation times is a bypassing of the update of the reluctance curl-curl matrix in conducting regions. The problem dynamics resulting from typical excitation currents are usually slow enough with the result that the small stable time step size results in an extreme oversampling. Thus, the changes in reluctivity in succeeding time steps are negligibly small and the update of the reluctance curl-curl matrix in the conducting regions is not required in every time step, if changes in reluctivity since the last update are noticeable. This bypassing strategy results in a significant computational speed-up, depending on the dynamics of the excitation.

In future research, this approach should also be evaluated for use with pulse-width-modulated excitation currents of fast dynamics. The pulse-width-modulated excitation currents are state of the art in motor control and thus of practical relevance. The small pulse width might result in the stable time step sizes just fulfilling the Nyquist criterion which will probably result in few to none bypassable matrix updates. Furthermore, within the work of this thesis, only the L_2 -norm has been used for computing changes in the solution vectors over time. In future research, other norms, such as e.g. the energy norm, should also be investigated as they might be more suitable from a physical point of view.

Another approach is the addition of a small offset value to the main diagonal elements of the singular curl-curl operator in nonconducting regions resulting in a regularization. The GPU-accelerated LU-decomposition of the mass matrix and of the regularized curl-curl operator in nonconducting regions was computed once and reused in each time step. The limit for this approach reached within the work of this thesis was

caused by matrix dimensions. Since only one GPU was used, the maximum matrix dimension was limited by the available memory of the GPU. This problem can be overcome in future works by using multiple GPUs, if the limiting communication bandwidth between the GPUs can be compensated for by a suitable algorithm.

With respect to all approaches presented, it should be mentioned that they were validated numerically using the ferromagnetic three-dimensional TEAM 10 benchmark problem with a rather coarse mesh. Since this strongly nonlinear magnetoquasistatic problem is suitable as an exemplary proof of concept of the proposed approaches.

Some speed-up can also certainly be gained by diagonalization of the mass matrix using the approach presented in [60], since a diagonal matrix inverse can be efficiently inverted and multiplied without using GPU-RAM.

Furthermore, time integration methods combining implicit and explicit time integration schemes, so called ImEx-methods should be subject to further investigations. Here, especially the Parareal algorithm combined with an ImEx-approach is of interest. On the coarse time grid, an implicit time integration method with an unconditionally stable time step size can be used. On the coarse time grid, only few time steps are required, thus only few computationally expensive implicit time steps are required. On the fine time grid, an explicit time integration scheme can be used. Here, many time steps are required. This corresponds well with computationally inexpensive, small, stable time steps lengths of explicit time integration schemes. In this context, the use of multiple GPUs is of interest, where each GPU can carry-out the computations in one coarse grid time interval.

Another topic covered in this thesis is the use of the Parareal algorithm for parallel-in-time integration of transient electroquasistatic field problems. Here, two implementations of the Parareal algorithm were investigated. In one implementation, the implicit Euler scheme was used for time integration on both the coarse and the fine time grid. In the other implementation the implicit Euler method was used on the coarse time grid and the explicit Euler scheme was used on the fine time grid. A simplified two-dimensional finite element model of a surge arrester with two differently fine meshes was used for numerical validation purposes. It was shown that the use of both Parareal implementations resulted in a computational speed-up, where the fastest simulations employed the explicit Euler method on the fine time grid. The work presented is only

to be considered a proof-of-concept for the applicability of the Parareal algorithm to electroquasistatic field problems. Further investigations should consider nonlinear field grading materials and three-dimensional problems. First simulations with a nonlinear field grading material were executed, but clearly show the need for further improvement, see e.g. [48]. The simulations employing the implicit Euler method on both the coarse and the fine time grid had a significantly larger simulation time than expected. Improving the linearization method employed might improve this. The simulations comprising a nonlinear field grading material and employing the Parareal algorithm with the implicit Euler method on the coarse time grid and the explicit Euler scheme on the fine time grid became numerically unstable.

Furthermore, simulations employing the Parareal algorithm and the explicit Euler method for time integration on both the coarse and the fine time grid and a linear field grading material were performed. These simulations also became numerically unstable and need to be investigated further. Here, employing a stabilized explicit time integration method, as e.g. the Runge-Kutta-Chebyshev method [103] with a larger stability region is of special interest.

Additionally, since the iterative nature of the Parareal algorithm comprises the repeated solving of multiple-right-hand side problem, the multiple-right-hand side techniques suggested for improved start vector generation for the conjugate gradient method in the magnetoquasistatic section of this thesis should also be applied. This will probably result in further speed-up of computations.

The Parareal algorithm also constitutes a good starting point for investigating a space-time-parallel approach. Here, the spatial part of the problem comprising only materials with linear or even constant material parameters are treated only on the coarse time grid. The spatial parts of the problem comprising nonlinear material properties are also treated on the fine time grid, since small time steps are more suitable to nonlinear problems.

Finally, it can be summarized that the work presented in this thesis, both with respect to the approaches suggested for the magnetoquasistatic and electroquasistatic problems, only represent ground work and proofs of concept. Yet, the approaches presented also open-up the possibility to employ further ideas, both mathematical and computing, respectively hardware-oriented.

Bibliography

- [1] R. C. Aiken, ed. *Stiff computation*. Oxford University Press, 1985.
- [2] S. Ausserhofer, O. Bíró, and K. Preis. “Discontinuous Galerkin finite elements in time domain eddy-current problems”. In: *IEEE Transactions on Magnetics* 45.3 (2009), pp. 1300–1303. ISSN: 00189464. DOI: 10.1109/TMAG.2009.2012604.
- [3] S. Außerhofer, O. Bíró, and K. Preis. “Discontinuous Galerkin formulation for eddy-current problems”. In: *COMPEL - The International Journal for Computation and Mathematics in Electrical and Electronic Engineering* 28.4 (2009), pp. 1081–1090. ISSN: 03321649. DOI: 10.1108/03321640910959125.
- [4] I. Babuška. “The finite element method with Langrangian multipliers”. In: *Numerische Mathematik* (1973). ISSN: 0029599X. DOI: 10.1007/BF01436561.
- [5] M. Bebendorf. “Adaptive Cross Approximation of Multivariate Functions”. In: *Constr Approx* 34 (2011), pp. 149–179.
- [6] O. Bíró, K. Preis, W. Renhart, K. R. Richter, and G. Vrisk. “Performance of different vector potential formulations in solving multiply connected 3-d eddy current problems”. In: *IEEE Transactions on Magnetics* 26.2 (1990), pp. 438–441. ISSN: 19410069. DOI: 10.1109/20.106347.
- [7] O. Bíró, K. Preis, and K. R. Richter. “Various FEM formulations for the calculation of transient 3d eddy currents in nonlinear media”. In: *IEEE Transactions on Magnetics* 31.3 (1995), pp. 1307–1312. ISSN: 19410069. DOI: 10.1109/20.376269.
- [8] W. Bosma, J. Cannon, and C. Playoust. “The Magma algebra system I: The user language”. In: *Journal of Symbolic Computation* (1997). ISSN: 07477171. DOI: 10.1006/jscs.1996.0125.

- [9] W. L. Briggs, Van E. Henson, and S. F. McCormick. *A Multigrid Tutorial, Second Edition*. Philadelphia, PA: SIAM, 2000. DOI: 10.1137/1.9780898719505.
- [10] J. C. Butcher. *Numerical Methods for Ordinary Differential Equations*. 2nd ed. Chichester, West Sussex: John Wiley & Sons, Ltd., 2008.
- [11] P. Castillo, R. Rieben, and D. White. “FEMSTER: An object oriented class library of discrete differential forms”. In: *IEEE Antennas and Propagation Society, AP-S International Symposium (Digest)*. 2003.
- [12] P. Castillo, R. Rieben, and D. White. *FEMSTER: An object-oriented class library of high-order discrete differential forms*. 2005. DOI: 10.1109/aps.2003.1219397.
- [13] M. Clemens, M. Helias, T. Steinmetz, and G. Wimmer. “Multiple right-hand side techniques for the numerical simulation of quasistatic electric and magnetic fields”. In: *Journal of Computational and Applied Mathematics* (2008). ISSN: 03770427. DOI: 10.1016/j.cam.2006.04.072.
- [14] M. Clemens, S. Schöps, H. De Gersem, and A. Bartel. “Decomposition and regularization of nonlinear anisotropic curl-curl DAEs”. In: *COMPEL - The International Journal for Computation and Mathematics in Electrical and Electronic Engineering* 30.6 (2011), pp. 1701–1714. ISSN: 03321649. DOI: 10.1108/03321641111168039.
- [15] M. Clemens, T. Steinmetz, and D. Weida. “Numerische Simulation elektroquasistatischer Hochspannungsfelder mit MEQSICO”. In: *Forschungsmagazin der Helmut-Schmidt-Universität, 17. Jahrgang* (2007), pp. 8–15.
- [16] M. Clemens and T. Weiland. “Transient eddy-current calculation with the FI-method”. In: *IEEE Transactions on Magnetics* (1999). ISSN: 00189464.
- [17] M. Clemens, M. Wilke, R. Schuhmann, and T. Weiland. “Subspace projection extrapolation scheme for transient field simulations”. In: *IEEE Transactions on Magnetics*. 2004. DOI: 10.1109/TMAG.2004.824583.
- [19] R. Courant, K. Friedrichs, and H. Lewy. “Über die partiellen Differenzengleichungen der mathematischen Physik”. In: *Mathematische Annalen* (1928).
- [20] C. F. Curtiss and J. O. Hirschfelder. “Integration of Stiff Equations”. In: *Proceedings of the National Academy of Sciences* 38 (1952), pp. 235–243. ISSN: 0027-8424. DOI: 10.1073/pnas.38.3.235.

-
- [21] J. Demmel, L. Grigori, M. Hoemmen, and J. Langou. “Communication-optimal parallel and sequential QR and LU factorizations”. In: *SIAM Journal on Scientific Computing* (2012). ISSN: 10648275. DOI: 10.1137/080731992. arXiv: 0808.2664.
- [22] P. Deuffhard and A. Hohmann. *Numerische Mathematik 1*. 4th ed. Berlin, New York: Walter de Gruyter, 2008.
- [23] H. K. Dirks. “Quasi-stationary fields for microelectronic applications”. In: *Springer Electrical Engineering* 79.2 (1996), pp. 145–155. ISSN: 0003-9039. DOI: 10.1007/BF01232924.
- [24] J. S. Dutiné, M. Clemens, and S. Schöps. “Explicit time integration of eddy current problems using a selective matrix update strategy”. In: *COMPEL - The International Journal for Computation and Mathematics in Electrical and Electronic Engineering* 36.5 (2017). ISSN: 03321649. DOI: 10.1108/COMPEL-02-2017-0100.
- [25] J. Dutiné, M. Clemens, C. Richter, S. Schöps, and G. Wimmer. “Explicit Time Integration of Eddy Current Problems with a Weakly Gauged Schur-Complement Vector Potential Formulation”. In: *URSI e.V. Kleinheubacher Tagung, Miltenberg, 28.-30.09.2015, Book of Abstracts*. 2015.
- [26] J. Dutiné, M. Clemens, and S. Schöps. “Explicit Time Integration of Eddy Current Problems Using a Selective Matrix Update Strategy”. In: *17th International IGTE Symposium on Numerical Field Calculation in Electrical Engineering (IGTE 2016), Graz, Austria, 18.-21.09.2016. Book of Abstracts*. 2016, p. 56.
- [27] J. Dutiné, M. Clemens, and S. Schöps. “GPU-Accelerated Explicit Time Integration of Eddy Current Problems”. In: *Computing and Electromagnetics International Workshop (CEM'17), Barcelona, Spain, 21.-24.06.2017*. 2017.
- [28] J. Dutiné, M. Clemens, and S. Schöps. “Multiple right-hand side techniques in semi-explicit time integration methods for transient eddy current problems”. In: *IEEE CEFC 2016 - 17th Biennial Conference on Electromagnetic Field Computation*. 2017. ISBN: 9781509010325. DOI: 10.1109/CEFC.2016.7815901.

- [29] J. Duti n , M. Clemens, and S. Sch ops. “Multiple Right-Hand Side Techniques in Semi-Explicit Time Integration Methods for Transient Eddy-Current Problems”. In: *IEEE Transactions on Magnetics* 53.6 (2017). ISSN: 00189464. DOI: 10.1109/TMAG.2017.2682558.
- [30] J. Duti n , M. Clemens, and S. Sch ops. “Numerical Techniques for the Acceleration of Semi-Explicit Time Integration of Eddy Current Problems”. In: *URSI e.V. Kleinheubacher Tagung, Miltenberg, 26.-28.09.2016, Book of Abstracts*. 2016.
- [31] J. Duti n , M. Clemens, and S. Sch ops. “Survey on Semi-Explicit Time Integration of Eddy Current Problems”. In: *Proceedings of the 11th International Conference on Scientific Computing in Electrical Engineering (SCEE 2016), Strobl, Austria, 03.-07.10.2016*. 2016.
- [32] J. Duti n , M. Clemens, S. Sch ops, and G. Wimmer. “Explicit time integration of transient eddy current problems”. In: *International Journal of Numerical Modelling: Electronic Networks, Devices and Fields* 31.2 (2018). ISSN: 10991204. DOI: 10.1002/jnm.2227.
- [33] J. Duti n , C. Richter, C. J rgens, S. Sch ops, and M. Clemens. “Explicit time integration techniques for electro- and magneto-quasistatic field simulations”. In: *Proceedings of the 2017 19th International Conference on Electromagnetics in Advanced Applications, ICEAA 2017*. 2017. ISBN: 9781509044511. DOI: 10.1109/ICEAA.2017.8065562.
- [34] L. Egiziano, V. Tucci, C. Petrarca, and M. Vitelli. “A Galerkin model to study the field distribution in electrical components employing nonlinear stress grading materials”. In: *IEEE Transactions on Dielectrics and Electrical Insulation* (1999). ISSN: 10709878. DOI: 10.1109/94.822012.
- [35] F. Fritzen, B. Haasdonk, D. Ryckelynck, and S. Sch ops. “An Algorithmic Comparison of the Hyper-Reduction and the Discrete Empirical Interpolation Method for a Nonlinear Thermal Problem”. In: *Mathematical and Computational Applications* (2018). DOI: 10.3390/mca23010008. arXiv: 1610.05029.
- [36] K. Fujiwara and T. Nakata. “Results for Benchmark Problem 7 (Asymmetrical Conductor with a Hole)”. In: *COMPEL - The International Journal for Computation and Mathematics in Electrical and Electronic Engineering* 9.3 (1990), pp. 137–154.

-
- [37] K. Fujiwara, T. Nakata, and H. Fusayasu. “Acceleration of Convergence Characteristic of the ICCG Method”. In: *IEEE Transactions on Magnetics* (1993). ISSN: 19410069. DOI: 10.1109/20.250792.
- [38] K. Gallivan, W. Jalby, Ul. Meier, and A. H. Sameh. “Impact of hierarchical memory systems on linear algebra algorithm design”. In: *International Journal of High Performance Computing Applications* (1988). ISSN: 17412846. DOI: 10.1177/109434208800200103.
- [39] M. J. Gander. “50 Years of Time Parallel Time Integration”. In: *Multiple Shooting and Time Domain Decomposition Methods*. Berlin: Springer, 2015, pp. 69–113. DOI: 10.1007/978-3-319-23321-5_3.
- [40] M. J. Gander and E. Hairer. “Nonlinear convergence analysis for the parareal algorithm”. In: *Domain Decomposition Methods in Science and Engineering XVII. Lecture Notes in Computational Science and Engineering, vol 60*. Ed. by U. Langer, M. Discacciati, D.E. Keyes, O.B. Widlund, and W. Zulehner. Berlin, Heidelberg: Springer, 2008, pp. 45–56. ISBN: 9783540751984. DOI: 10.1007/978-3-540-75199-1_4.
- [41] M. J. Gander and S. Vandewalle. “Analysis of the parareal time-parallel time-integration method”. In: *SIAM Journal on Scientific Computing* (2007). ISSN: 10648275. DOI: 10.1137/05064607X.
- [42] M. Gee, C. Siefert, J. Hu, R. Tuminaro, and M. Sala. “ML 5.0 smoothed aggregation user’s guide”. In: *Sandia National Laboratories* (2006).
- [43] G. H. Golub and C. F. Van Loan. *Matrix Computations*. 4th ed. Baltimore: The John Hopkins University Press, 2013.
- [44] L. Grigori, J. W. Demmel, and H. Xiang. “CALU: A communication optimal LU factorization algorithm”. In: *SIAM Journal on Matrix Analysis and Applications* (2011). ISSN: 08954798. DOI: 10.1137/100788926.
- [45] E. Hairer and G. Wanner. *Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems*. 2nd ed. Berlin: Springer-Verlag Berlin Heidelberg, 1996. ISBN: 978-3-540-60452-5. DOI: 10.1007/978-3-642-05221-7.
- [46] O. Heaviside. *Electromagnetic theory, vol. I*. London, 1893.
- [47] O. Heaviside. *Electromagnetic theory, vol. II*. London, 1899.

- [48] M. Henkel, F. Kasolis, and M. Clemens. “Parallel-in-Time Simulation of Nonlinear Transient Electro-Quasistatic Field Problems”. In: *22nd Conference on the Computation of Electromagnetic Fields (COMPUMAG 2019), Paris, France, 15.-19.07.2019. Extended Abstract. Full paper accepted for publication in IEEE-Explore*. 2019.
- [49] T. Henneron and S. Clénet. “Model order reduction of non-linear magneto-static problems based on POD and DEI methods”. In: *IEEE Transactions on Magnetism* (2014). ISSN: 00189464. DOI: 10.1109/TMAG.2013.2283141.
- [50] T. Henneron and S. Clénet. “Model-order reduction of multiple-input non-linear systems based on POD and DEI methods”. In: *IEEE Transactions on Magnetism* (2015). ISSN: 00189464. DOI: 10.1109/TMAG.2014.2347416.
- [51] M. A. Heroux, R. A. Bartlett, V. E. Howle, R. J. Hoekstra, J. J. Hu, T. G. Kolda, R. B. Lehoucq, K. R. Long, R. P. Pawlowski, E. T. Phipps, A. G. Salinger, H. K. Thornquist, R. S. Tuminaro, J. M. Willenbring, A. Williams, and K. S. Stanley. *An overview of the trilinos project*. 2005. DOI: 10.1145/1089014.1089021.
- [52] R. Hiptmair. “Finite elements in computational electromagnetism”. In: *Acta Numerica* (2002). ISSN: 0962-4929. DOI: 10.1017/s0962492902000041.
- [53] R. Hiptmair. “Multigrid method for Maxwell’s equations”. In: *SIAM Journal on Numerical Analysis* (1999). ISSN: 00361429. DOI: 10.1137/s0036142997326203.
- [54] R. Holland. “FDTD Analysis of Nonlinear Magnetic Diffusion by Reduced c”. In: *IEEE Transactions on Antennas and Propagation* 43.7 (1995), pp. 653–659. ISSN: 15582221. DOI: 10.1109/8.391135.
- [55] International Electrotechnical Commission. “Surge arresters - Part 4: Metal-oxide surge arresters without gaps for a.c. systems”. In: *IEC 60099-4* (2009).
- [56] J.-M. Jin. *The Finite Element Method in Electromagnetics*. 3rd ed. Hoboken, New Jersey: John Wiley & Sons, Inc., 2014, p. 878.
- [57] A. Kameari. “Calculation of transient 3d eddy current using edge-elements”. In: *IEEE Transactions on Magnetism* (1990). ISSN: 19410069. DOI: 10.1109/20.106354.

-
- [58] F. Kasolis and M. Clemens. “Information-based model reduction for nonlinear electro-quasistatic problems”. In: *Journal of Computational Physics* (2020). ISSN: 10902716. DOI: 10.1016/j.jcp.2019.109118.
- [59] G. D. Knott. *Interpolating Cubic Splines*. Boston, MA: Birkhäuser Boston, 2000.
- [60] P. Lacoste. “La condensation de la matrice masse, ou mass-lumping, pour les éléments finis mixtes de Raviart-Thomas-Nédélec d’ordre 1”. In: *Comptes Rendus Mathématique* (2004). ISSN: 1631073X. DOI: 10.1016/j.crma.2004.09.013.
- [61] M. G. Larson and F. Bengzon. *The Finite Element Method: Theory, Implementation, and Applications*. 1st ed. Berlin et al.: Springer-Verlag Berlin Heidelberg, 2013.
- [62] J. L. Lions, Y. Maday, and G. Turinici. “A "parareal" in time discretization of PDE’s”. In: *Comptes Rendus de l’Académie des Sciences - Series I: Mathematics* 332 (2001), pp. 661–668. ISSN: 0764-4442. DOI: 10.1016/S0764-4442(00)01793-6.
- [63] K. R. Long, R. S. Tuminaro, R. A. Bartlett, R. J. Hoekstra, E. T. Phipps, T. G. Kolda, R. B. Lehoucq, H. K. Thornquist, J. J. Hu, A. B. Williams, A. G. Salinger, Vi. E. Howle, R. P. Pawlowski, J. M. Willenbring, and M. A. Heroux. *An overview of Trilinos*. Tech. rep. Albuquerque, NM, and Livermore, CA: Sandia National Laboratories (SNL), Aug. 2003. DOI: 10.2172/918383. URL: <http://www.osti.gov/servlets/purl/918383-cIoGXQ/>.
- [64] N. Lord, G. H. Golub, and C. F. Van Loan. “Matrix Computations”. In: *The Mathematical Gazette* 83.498 (1999), p. 556. ISSN: 00255572. DOI: 10.2307/3621013.
- [65] R. Luebbers, K. Kumagai, S. Adachi, and T. Uno. “FDTD Calculation of Transient Pulse Propagation through a Nonlinear Magnetic Sheet”. In: *IEEE Transactions on Electromagnetic Compatibility* 35.1 (1993), pp. 90–94. ISSN: 1558187X. DOI: 10.1109/15.249400.
- [66] J. B. Manges and Z. J. Cendes. “A Generalized Tree-Cotree Gauge for Magnetic Field Computation”. In: *IEEE Transactions on Magnetics* (1995). ISSN: 19410069. DOI: 10.1109/20.376275.

- [67] *MATLAB Release 2017a*. Natick, Massachusetts, United States.
- [68] J. C. Maxwell. *A Treatise on Electricity and Magnetism*. Oxford: Clarendon Press, 1873.
- [69] P. Monk. *Finite Element Methods for Maxwell's Equations*. Oxford: Oxford University Press, 2003, p. 464.
- [70] T. Nakata and K. Fujiwara. “Results for Benchmark Problem 10 (Steel Plates Around a Coil)”. In: *COMPEL - The International Journal for Computation and Mathematics in Electrical and Electronic Engineering* 9.3 (1990), pp. 181–1990.
- [71] T. Nakata, N. Takahashi, and K. Fujiwara. *Summary of results for benchmark problem 10 (steel plates around a coil)*. 1995. DOI: 10.1108/eb010141.
- [72] A. Nicolet. “Implicit runge-kutta methods for transient magnetic field computation”. In: *IEEE Transactions on Magnetics* (1996). ISSN: 00189464. DOI: 10.1109/20.497510.
- [73] J. Nievergelt. “Parallel methods for integrating ordinary differential equations”. In: *Communications of the ACM* (1964). ISSN: 15577317. DOI: 10.1145/355588.365137.
- [74] K. Preis. “On the use of the magnetic vector potential in the finite-element analysis of three-dimensional eddy - Magnetics, IEEE Transactions on”. In: *Potentials* 25.4 (1989).
- [75] K. Preis, O. Biro, W. Renhart, K. R. Richter, and G. Vrisk. “Various FEM formulations for the calculation of transient 3d eddy currents in nonlinear media”. In: *IEEE Transactions on Magnetics* 26.3 (1995), pp. 1307–1312. ISSN: 19410069. DOI: 10.1109/20.376269.
- [76] S. Reitzinger. “Algebraic Multigrid Methods for Large Scale Finite Element Equations”. PhD thesis. Johannes Kepler Universität Linz, 2001.
- [77] S. Reitzinger and J. Schöberl. “An algebraic multigrid method for finite element discretizations with edge elements”. In: *Numerical Linear Algebra with Applications* (2002). ISSN: 10705325. DOI: 10.1002/nla.271.
- [78] C. Richter. “Beiträge zur Beschleunigung numerischer Simulationen von elektroquasistatischen Feldverteilungen”. PhD thesis. Bergische Universität Wuppertal, 2019.

-
- [79] C. Richter, S. Schöps, and M. Clemens. “GPU Accelerated Explicit Time-Integration Methods for Electro-Quasi-Static Fields”. In: *IEEE Transactions on Magnetics* (2017). ISSN: 00189464. DOI: 10.1109/TMAG.2017.2662234.
- [80] C. Richter, S. Schöps, and M. Clemens. “GPU acceleration of algebraic multi-grid preconditioners for discrete elliptic field problems”. In: *IEEE Transactions on Magnetics* (2014). ISSN: 00189464. DOI: 10.1109/TMAG.2013.2283099.
- [81] C. Richter, S. Schöps, and M. Clemens. “Multi-GPU acceleration of algebraic multi-grid preconditioners for elliptic field problems”. In: *IEEE Transactions on Magnetics* (2015). ISSN: 00189464. DOI: 10.1109/TMAG.2014.2357332.
- [82] C. Richter, S. Schöps, and M. Clemens. “Multi-GPU Acceleration of Algebraic Multigrid Preconditioners”. In: 2016. DOI: 10.1007/978-3-319-30399-4_9.
- [84] Y. Robert. “The Impact of Vector and Parallel Architectures on the Gaussian Elimination Algorithm”. In: *The Academy of Management Review* (1991). ISSN: 03637425. DOI: 10.2307/257255.
- [85] T. Rylander, P. Ingelström, and A. Bonderson. *Computational Electromagnetics*. 2nd ed. Berlin et al.: Springer-Verlag Berlin Heidelberg, 2013.
- [86] Y. Saad. *Iterative Methods for Sparse Linear Systems*. 2nd ed. Philadelphia, PA: SIAM, 2003.
- [87] E. N. Sarmin and L. A. Chudov. “On the stability of the numerical integration of systems of ordinary differential equations arising in the use of the straight line method”. In: *USSR Computational Mathematics and Mathematical Physics* (1963). ISSN: 00415553. DOI: 10.1016/0041-5553(63)90256-8.
- [88] Y. Sato, M. Clemens, and H. Igarashi. “Adaptive Subdomain Model Order Reduction with Discrete Empirical Interpolation Method for Nonlinear Magneto-Quasi-Static Problems”. In: *IEEE Transactions on Magnetics*. 2016. DOI: 10.1109/TMAG.2015.2489264.
- [89] O. Schenk. “Scalable parallel sparse LU factorization methods on shared memory multiprocessors”. PhD thesis. ETH Zürich, 2000.
- [90] W. E. Schiesser. *The numerical method of lines : integration of partial differential equations*. San Diego: Academic Press, Inc., 1991.

- [91] D. Schmidthäusler, S. Schöps, and M. Clemens. “Linear subspace reduction for quasistatic field simulations to accelerate repeated computations”. In: *IEEE Transactions on Magnetics* (2014). ISSN: 00189464. DOI: 10.1109/TMAG.2013.2280693.
- [92] S. Schöps, A. Bartel, and M. Clemens. “Higher Order Half-Explicit Time Integration of Eddy Current Problems Using Domain Substructuring”. In: *IEEE Transactions on Magnetics* 48.2 (Feb. 2012), pp. 623–626. ISSN: 0018-9464. DOI: 10.1109/TMAG.2011.2172780. URL: <http://ieeexplore.ieee.org/document/6136499/>.
- [93] S. Schöps, H. De Gersem, and T. Weiland. “Winding Functions in Transient Magnetoquasistatic Field-Circuit Coupled Simulations”. In: *The International Journal for Computation and Mathematics in Electrical and Electronic Engineering* 32.6 (2013), pp. 2063–2083.
- [94] S. Schöps, I. Niyonzima, and M. Clemens. “Parallel-In-Time Simulation of Eddy Current Problems Using Parareal”. In: *IEEE Transactions on Magnetics* 54.3 (Mar. 2018), pp. 1–4. ISSN: 0018-9464. DOI: 10.1109/TMAG.2017.2763090. arXiv: 1706.05750. URL: <http://ieeexplore.ieee.org/document/8093684/>.
- [95] J. R. Shewchuk. *An Introduction to the Conjugate Gradient Method Without the Agonizing Pain*. Tech. rep. Pittsburgh, PA: School of Computer Science, Carnegie Mellon University, 1994. URL: <http://www.cs.cmu.edu/~7B/~7Dquake-papers/painless-conjugate-gradient.pdf>.
- [96] P. P. Silvester and R. L. Ferrari. *Finite Elements for Electrical Engineers*. 3rd ed. Cambridge: Cambridge University Press, 1996.
- [97] J. Smajic. “A novel variant of the H- Φ field formulation for magnetostatic and eddy current problems”. In: *COMPEL - The International Journal for Computation and Mathematics in Electrical and Electronic Engineering* (2019). ISSN: 03321649. DOI: 10.1108/COMPEL-12-2018-0536.
- [98] J. Smajic, M. K. Bucher, C. Jager, and R. Christen. “Treatment of Multiply Connected Domains in Time-Domain Discontinuous Galerkin H- Φ Eddy Current Analysis”. In: *IEEE Transactions on Magnetics* (2019). ISSN: 19410069. DOI: 10.1109/TMAG.2018.2890525.

-
- [99] J. Smajic, M. Bucher, R. Christen, and Z. Tanasic. “DG-FEM for Time Domain H- Φ Eddy Current Analysis”. In: *IEEE Transactions on Magnetics* 53.6 (2017), pp. 1–4. ISSN: 00189464. DOI: 10.1109/TMAG.2017.2657654.
- [100] J. Smajic, M. Bucher, R. Christen, and Z. Tanasic. “Stability Analysis of Time Domain Discontinuous Galerkin H- Φ Method for Eddy Current Simulations”. In: *IEEE Transactions on Magnetics* 54.3 (2018), pp. 1–4. ISSN: 00189464. DOI: 10.1109/TMAG.2017.2743159.
- [101] I. M. Smith and D. V. Griffiths. *Programming the Finite Element Method*. 4th ed. Chichester, West Sussex: John Wiley & Sons, Ltd., 2004.
- [102] E. Solomonik and J. Demmel. “Communication-optimal parallel 2.5D matrix multiplication and LU factorization algorithms”. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. 2011. ISBN: 9783642233968. DOI: 10.1007/978-3-642-23397-5_10.
- [103] B. P. Sommeijer, L. F. Shampine, and J. G. Verwer. “RKC: An explicit solver for parabolic PDEs”. In: *Journal of Computational and Applied Mathematics* (1998). ISSN: 03770427. DOI: 10.1016/S0377-0427(97)00219-7.
- [104] T. Steinmetz, M. Helias, G. Wimmer, L.O. Fichte, and M. Clemens. “Electro-quasistatic field simulations based on a discrete electromagnetism formulation”. In: *IEEE Transactions on Magnetics* 42.4 (Apr. 2006), pp. 755–758. ISSN: 0018-9464. DOI: 10.1109/TMAG.2006.872488. URL: <http://ieeexplore.ieee.org/document/1608316/>.
- [105] E. Stiefel. “Über einige Methoden der Relaxationsrechnung”. In: *ZAMP Zeitschrift für angewandte Mathematik und Physik* (1952). ISSN: 00442275. DOI: 10.1007/BF02080981.
- [106] K. Stüben. *Algebraic multigrid (AMG): an introduction with applications*. GMD-Report 70, 1999.
- [107] S. Tomov, J. Dongarra, and M. Baboulin. “Towards dense linear algebra for hybrid GPU accelerated manycore systems”. In: *Parallel Computing* (2010). ISSN: 01678191. DOI: 10.1016/j.parco.2009.12.005.
- [108] L. N. Trefethen and D. Bau. *Numerical Linear Algebra*. Philadelphia: Society for Industrial and Applied Mathematics, 1997.

- [109] U. Trottenberg, C. Oosterlee, and A. Schuller. “Multigrid”. In: 1st ed. San Diego et al.: Academic Press, Inc., 2000.
- [110] J. C. Verite. “Calculation of Multivalued Potentials in Exterior Regions”. In: *IEEE Transactions on Magnetism* (1987). ISSN: 19410069. DOI: 10.1109/TMAG.1987.1065167.
- [111] J. G. Verwer. “Explicit Runge-Kutta methods for parabolic partial differential equations”. In: *Applied Numerical Mathematics* (1996). ISSN: 01689274. DOI: 10.1016/s0168-9274(96)00022-0.
- [112] T. Weiland. “Discretization Method for the Solution of Maxwell’s Equations for Six-Component Fields.” In: *AEU-Archiv fur Elektronik und Ubertragungstechnik* (1977). ISSN: 1434-8411.
- [113] T. Weiland. “Time domain electromagnetic field computation with finite difference methods”. In: *International Journal of Numerical Modelling: Electronic Networks, Devices and Fields* (1996). ISSN: 08943370. DOI: 10.1002/(SICI)1099-1204(199607)9:4<295::AID-JNM240>3.0.CO;2-8.
- [114] G. Wimmer, M. Clemens, and J. Lang. “Calculation of Magnetic Fields with Finite Elements”. In: *From Nano to Space*. Ed. by M. H. Breitner, G. Denk, and P. Rentrop. Berlin, Heidelberg: Springer, 2008, pp. 111–124.
- [115] K. S. Yee. “Numerical Solution of Initial Boundary Value Problems Involving Maxwell’s Equations in Isotropic Media”. In: *IEEE Transactions on Antennas and Propagation* (1966). ISSN: 15582221. DOI: 10.1109/TAP.1966.1138693.
- [116] T. V Yioultsis. “Explicit Finite-Difference Time-Domain Method With Perfectly Matched Layers for Transient”. In: *IEEE Transactions on Magnetism* 38.5 (2002), pp. 3439–3447.
- [117] T. V. Yioultsis, K. S. Charitou, C. S. Antonopoulos, and T. D. Tsiboukis. “A finite difference time domain scheme for transient eddy current problems”. In: *IEEE Transactions on Magnetism* 37.5 I (2001), pp. 3145–3149. ISSN: 00189464. DOI: 10.1109/20.952563.
- [120] O. C. Zienkiewicz and R. L. Taylor. “The Finite Element Method Volume 1 : The Basis”. In: *Methods* (2000).

Publications by the Author

- [18] M. Clemens, M. Zang, C. Cimala, J. Dutiné, J. Streckert, T. Timm, and B. Schmuelling. “Simulation der Exposition des menschlichen Körpers durch magneto-quasistatische Felder von induktiven Ladesystemen in Automobilen”. In: *Internationale Fachmesse und Kongress für Elektromagnetische Verträglichkeit (EMV 2018), Düsseldorf, 20.-22.02.2018, Germany. Tagungsband.* 2018, pp. 295–303.
- [24] J. S. Dutiné, M. Clemens, and S. Schöps. “Explicit time integration of eddy current problems using a selective matrix update strategy”. In: *COMPEL - The International Journal for Computation and Mathematics in Electrical and Electronic Engineering* 36.5 (2017). ISSN: 03321649. DOI: 10.1108/COMPEL-02-2017-0100.
- [25] J. Dutiné, M. Clemens, C. Richter, S. Schöps, and G. Wimmer. “Explicit Time Integration of Eddy Current Problems with a Weakly Gauged Schur-Complement Vector Potential Formulation”. In: *URSI e.V. Kleinheubacher Tagung, Miltenberg, 28.-30.09.2015, Book of Abstracts.* 2015.
- [26] J. Dutiné, M. Clemens, and S. Schöps. “Explicit Time Integration of Eddy Current Problems Using a Selective Matrix Update Strategy”. In: *17th International IGTE Symposium on Numerical Field Calculation in Electrical Engineering (IGTE 2016) , Graz, Austria, 18.-21.09.2016. Book of Abstracts.* 2016, p. 56.
- [27] J. Dutiné, M. Clemens, and S. Schöps. “GPU-Accelerated Explicit Time Integration of Eddy Current Problems”. In: *Computing and Electromagnetics International Workshop (CEM'17), Barcelona, Spain, 21.-24.06.2017.* 2017.

- [28] J. Dutiné, M. Clemens, and S. Schöps. “Multiple right-hand side techniques in semi-explicit time integration methods for transient eddy current problems”. In: *IEEE CEFC 2016 - 17th Biennial Conference on Electromagnetic Field Computation*. 2017. ISBN: 9781509010325. DOI: 10.1109/CEFC.2016.7815901.
- [29] J. Dutiné, M. Clemens, and S. Schöps. “Multiple Right-Hand Side Techniques in Semi-Explicit Time Integration Methods for Transient Eddy-Current Problems”. In: *IEEE Transactions on Magnetics* 53.6 (2017). ISSN: 00189464. DOI: 10.1109/TMAG.2017.2682558.
- [30] J. Dutiné, M. Clemens, and S. Schöps. “Numerical Techniques for the Acceleration of Semi-Explicit Time Integration of Eddy Current Problems”. In: *URSI e.V. Kleinheubacher Tagung, Miltenberg, 26.-28.09.2016, Book of Abstracts*. 2016.
- [31] J. Dutiné, M. Clemens, and S. Schöps. “Survey on Semi-Explicit Time Integration of Eddy Current Problems”. In: *Proceedings of the 11th International Conference on Scientific Computing in Electrical Engineering (SCEE 2016), Strobl, Austria, 03.-07.10.2016*. 2016.
- [32] J. Dutiné, M. Clemens, S. Schöps, and G. Wimmer. “Explicit time integration of transient eddy current problems”. In: *International Journal of Numerical Modelling: Electronic Networks, Devices and Fields* 31.2 (2018). ISSN: 10991204. DOI: 10.1002/jnm.2227.
- [33] J. Dutiné, C. Richter, C. Jörgens, S. Schöps, and M. Clemens. “Explicit time integration techniques for electro- and magneto-quasistatic field simulations”. In: *Proceedings of the 2017 19th International Conference on Electromagnetics in Advanced Applications, ICEAA 2017*. 2017. ISBN: 9781509044511. DOI: 10.1109/ICEAA.2017.8065562.
- [83] C. Richter, S. Schöps, J. Dutiné, R. Schreiber, and M. Clemens. “Transient Simulation of Nonlinear Electro-Quasi-Static Field Problems Accelerated by Multiple GPUs”. In: *IEEE Transactions on Magnetics*. Vol. 52. 3. 2016. DOI: 10.1109/TMAG.2015.2466602.
- [118] M. Zang, C. Cimala, M. Clemens, J. Dutiné, and B. Schmuelling. “A Co-simulation Scalar-Potential Finite Difference (SPFD) approach for the simulation of human exposure to magneto-quasistatic fields”. In: *IEEE CEFC 2016*

- *17th Biennial Conference on Electromagnetic Field Computation*. 2017. ISBN: 9781509010325. DOI: 10.1109/CEFC.2016.7816104.
- [119] M. Zang, C. Cimala, M. Clemens, J. Dutilleul, T. Timm, and B. Schmuelling. “A Co-Simulation Scalar-Potential Finite Difference Method for the Numerical Analysis of Human Exposure to Magneto-Quasi-Static Fields”. In: *IEEE Transactions on Magnetics* 53.6 (2017). ISSN: 00189464. DOI: 10.1109/TMAG.2017.2657819.

